FIG. 1

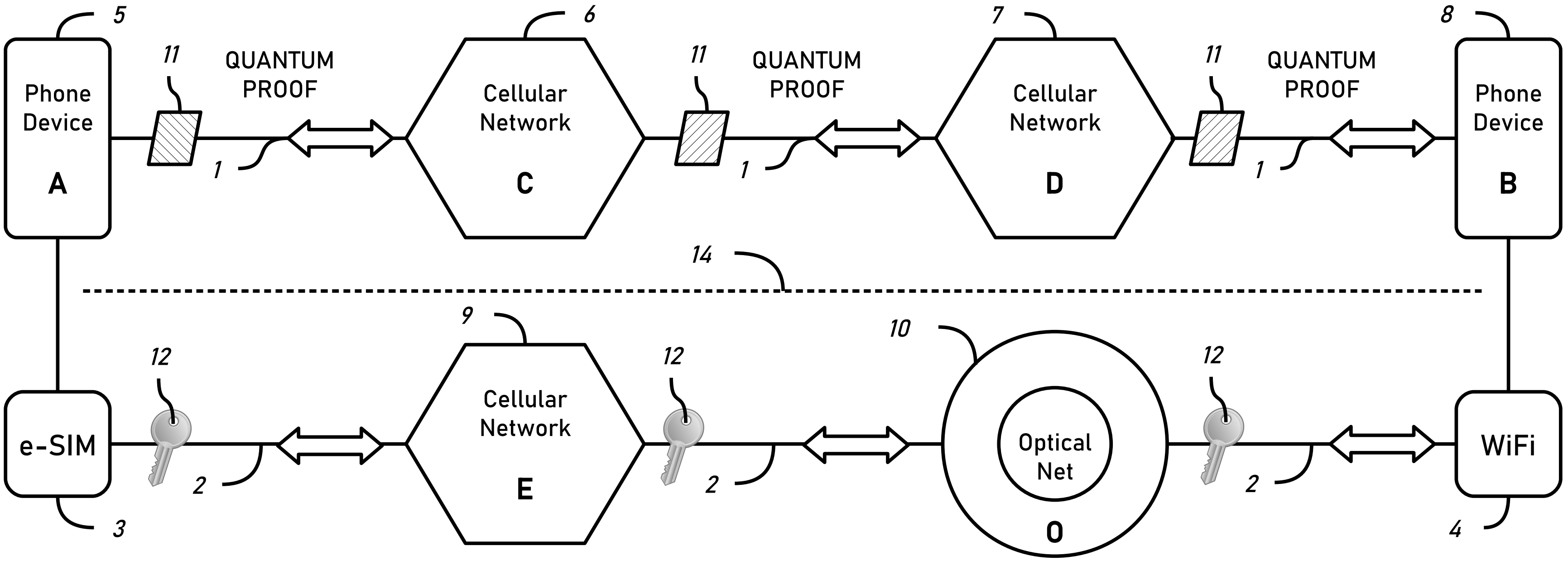

5
6
7
8
11
QUANTUM PROOF
11
QUANTUM PROOF
11
QUANTUM PROOF
Phone Device
A
Cellular Network
C
Cellular Network
D
Phone Device
B
1
1
1
14
9
10
12
12
12
e-SIM
Cellular Network
E
Optical Net
O
WiFi
2
2
2
3
4

FIG. 1A

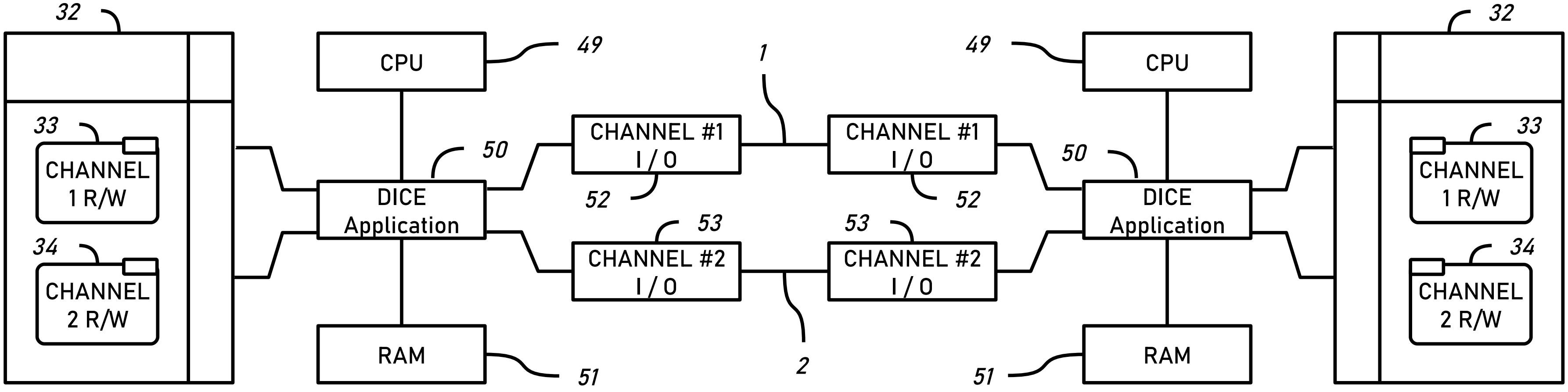

32
32
CPU
49
49
CPU
1
33
CHANNEL 1 R/W
CHANNEL #1 I / O
CHANNEL #1 I / O
33
CHANNEL 1 R/W
50
DICE Application
52
52
50
DICE Application
34
CHANNEL 2 R/W
53
CHANNEL #2 I / O
53
CHANNEL #2 I / O
34
CHANNEL 2 R/W
RAM
51
2
51
RAM

FIG. 2

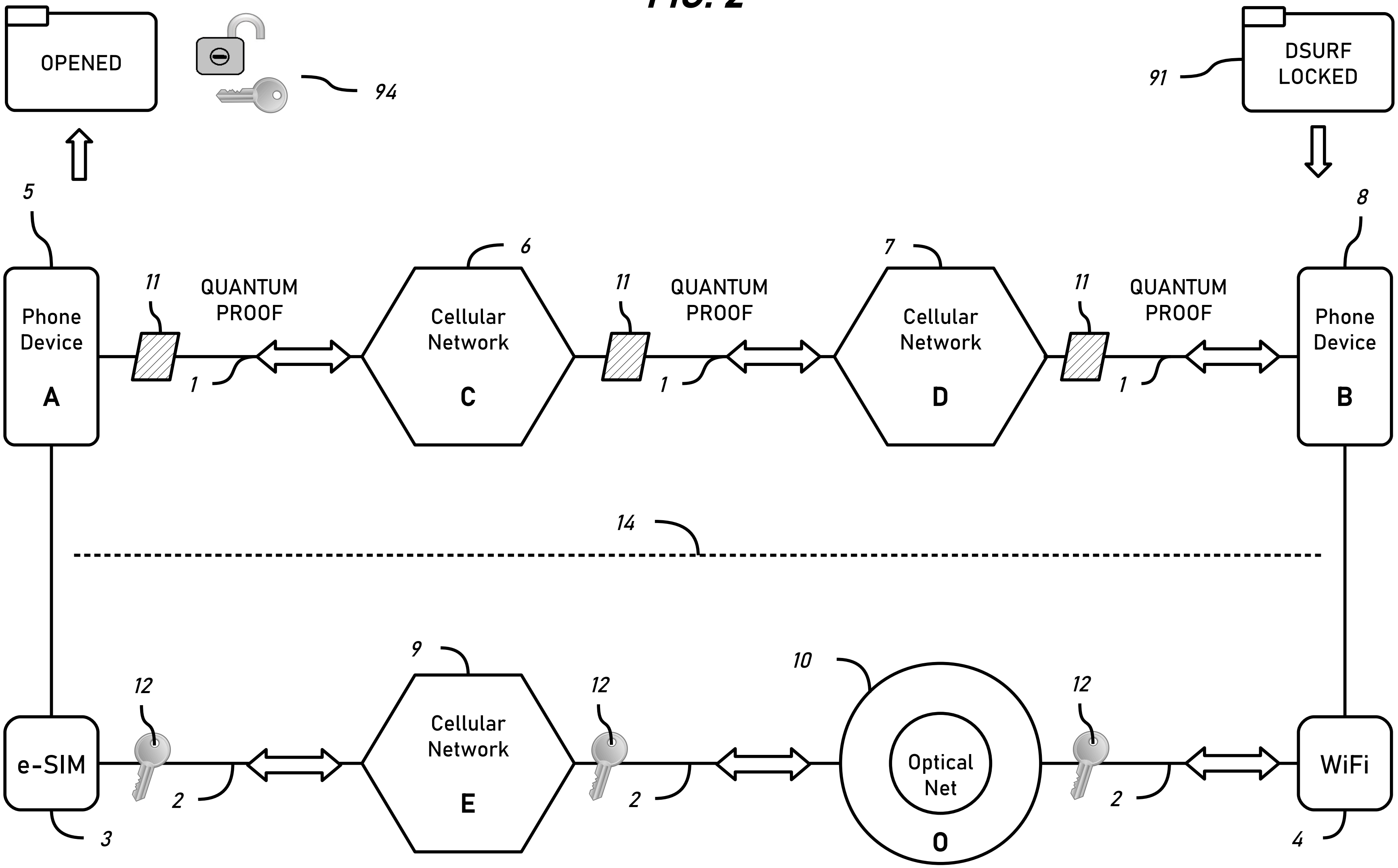

OPENED
94
DSURF
LOCKED
91
5
8
Phone
Device
A
11
QUANTUM
PROOF
1
6
Cellular
Network
C
11
QUANTUM
PROOF
1
7
Cellular
Network
D
11
QUANTUM
PROOF
1
Phone
Device
B
14
9
10
12
e-SIM
3
12
2
Cellular
Network
E
12
2
Optical
Net
O
12
2
WiFi
4

FIG. 3

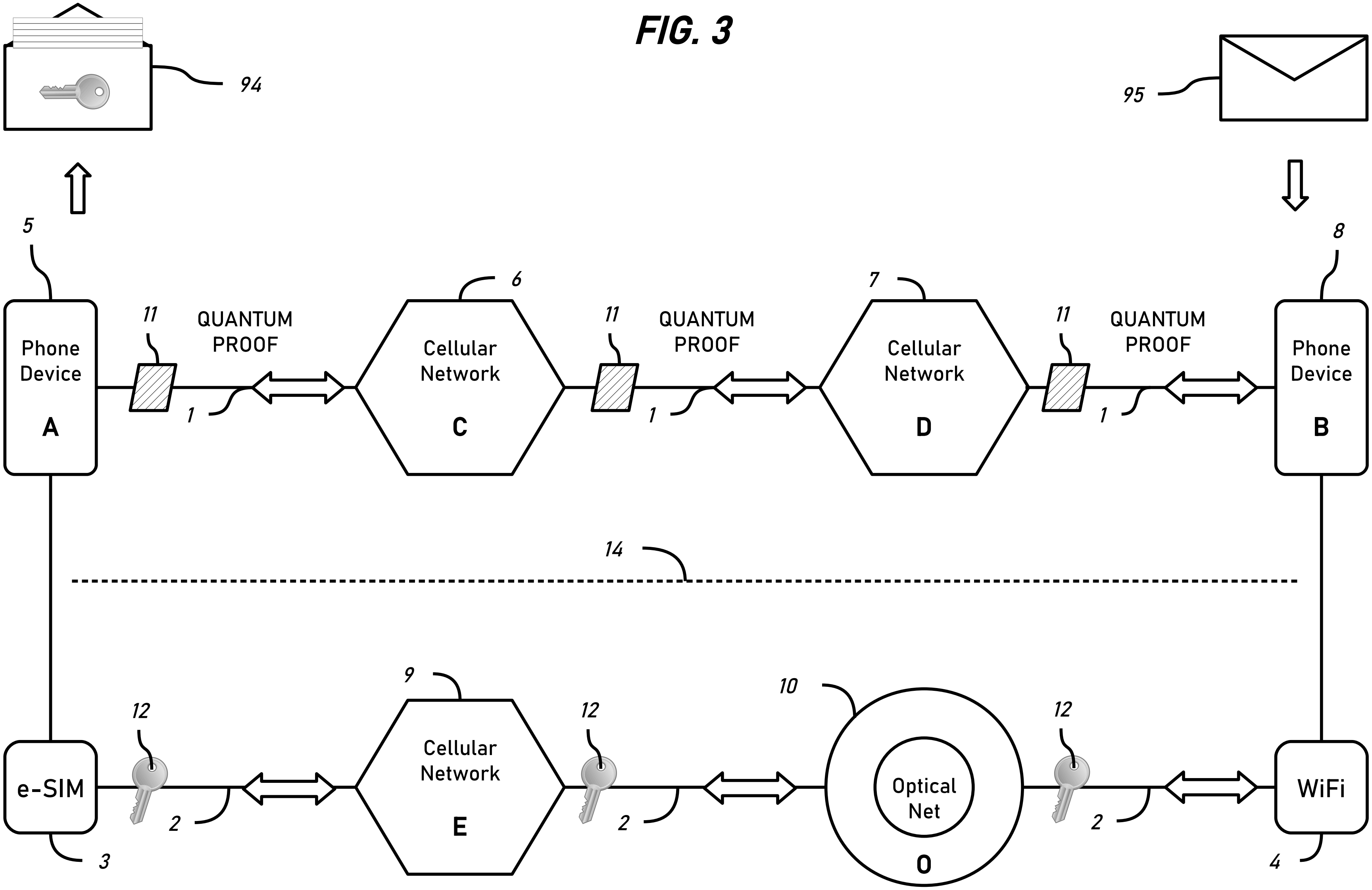

94
95
5
Phone Device
A
11
QUANTUM PROOF
1
6
Cellular Network
C
11
QUANTUM PROOF
1
7
Cellular Network
D
11
QUANTUM PROOF
1
8
Phone Device
B
14
e-SIM
3
12
2
9
Cellular Network
E
12
2
10
Optical Net
O
12
2
WiFi
4

FIG. 4

54 DEFINED PROCESS

DUAL ISOLATED CHANNELS ENCRYPTION

69. The local system and remote system may begin using any encryption such as HTTPS via web-browser or no encryption, operating through a telecommunications network infrastructure and / or the Internet.

70. Initiate contact with remote system through Channel 1.

71. Request dual isolated channels encryption.

72. If accepted exchange encryption keys, plus any rules and protocols:

    73. create and write to / or retrieve local system private key from Channel 2 storage (34)

    74. initiate connection to remote system through Channel 2

    75. send / or receive private key through Channel 2.

76. Start send and / or receive encrypted communications data exchange over Channel 1 using key(s) and any protocols stored in Channel 2 storage (34).

77. Receive / or provide remote system private key through Channel 1:

    78. write private key to Channel 1 storage (33)

    79. retrieve private key from Channel 1 storage (33).

80. Start send and / or receive encrypted communications data exchange over Channel 2 using key(s) and any protocols stored in Channel 1 storage (33).

81. Periodically refresh / generate and exchange new keys according to a parameter (that may be randomly timed or set to zero to switch off).

82. Retain keys in storage upon instances of dropped or lost connections.

83. Resume using stored keys or if corrupted loop back (→ 70).

84. Terminate at end of session – may include preset option to retain keys in storage (33, 34).

85. Resume using stored keys or if corrupted loop back (→ 70).

FIG. 4A

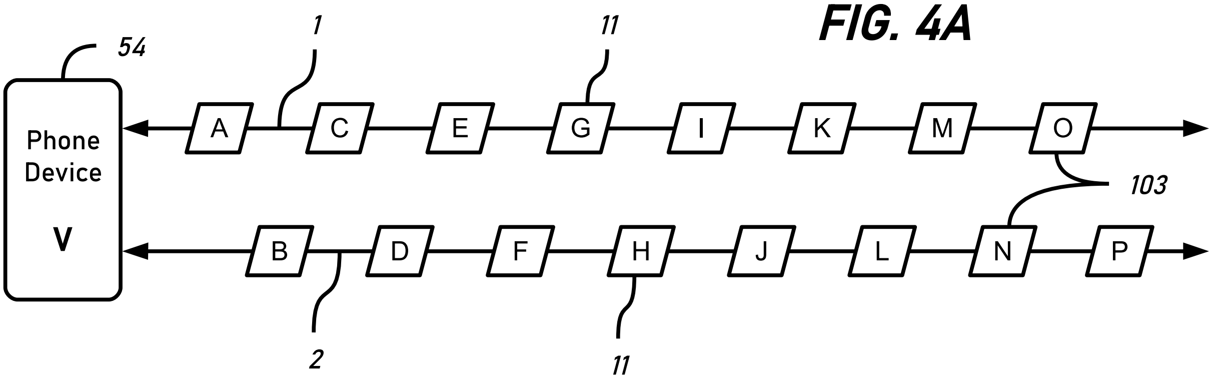


FIG. 4B

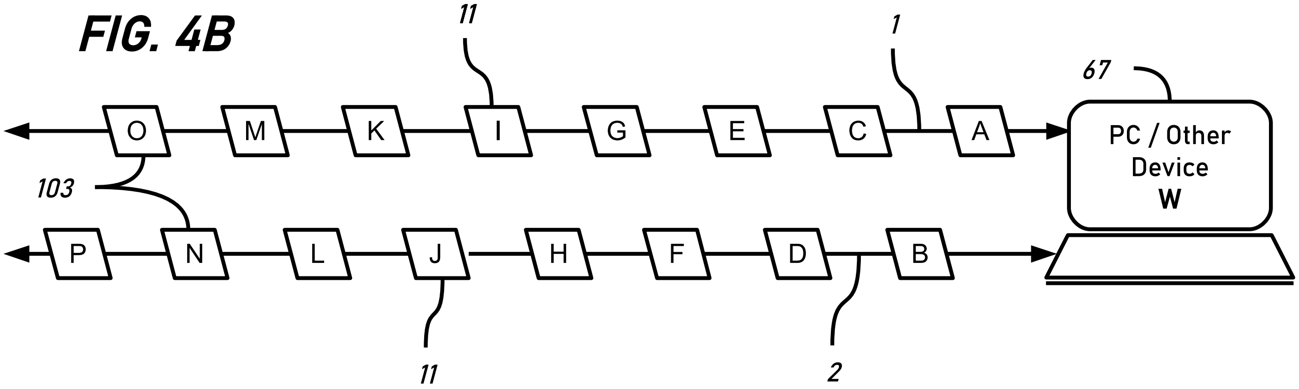


FIG. 4C

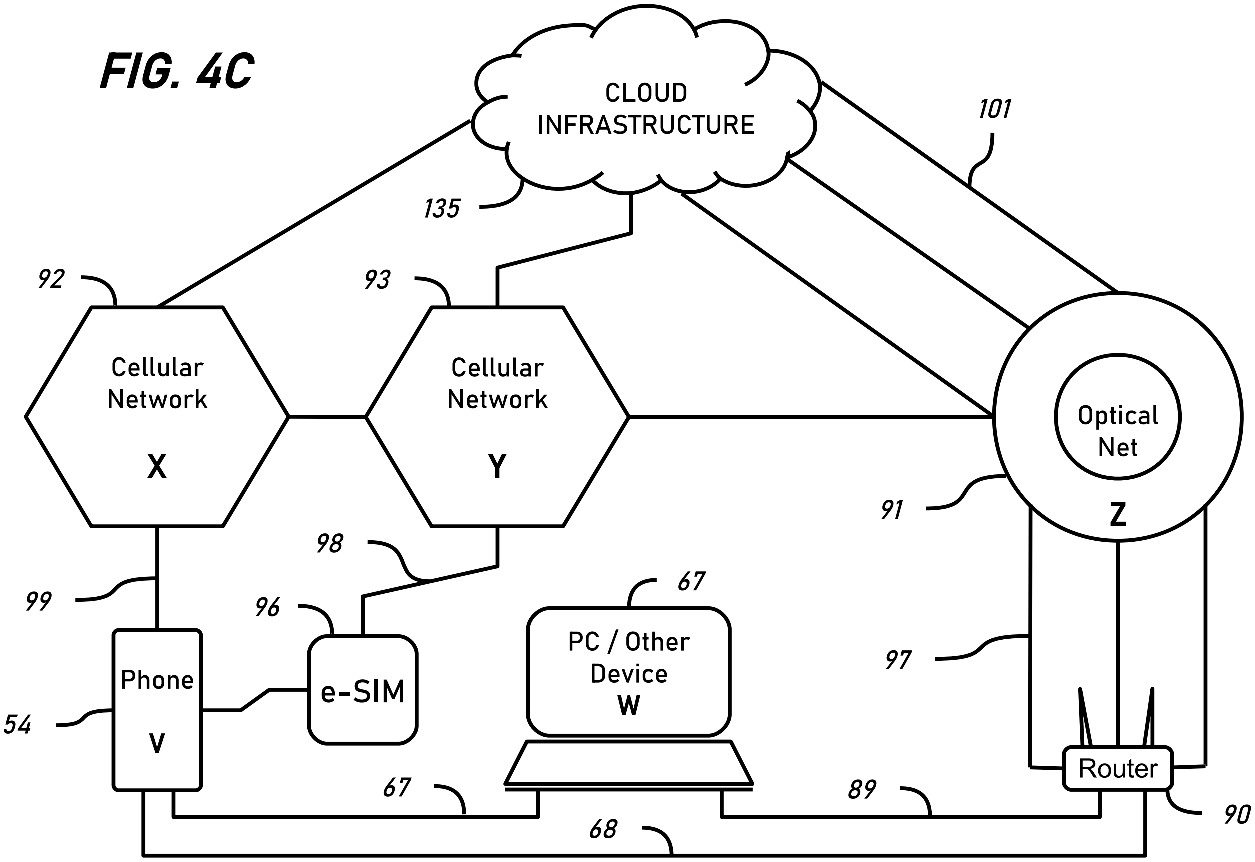


FIG. 5

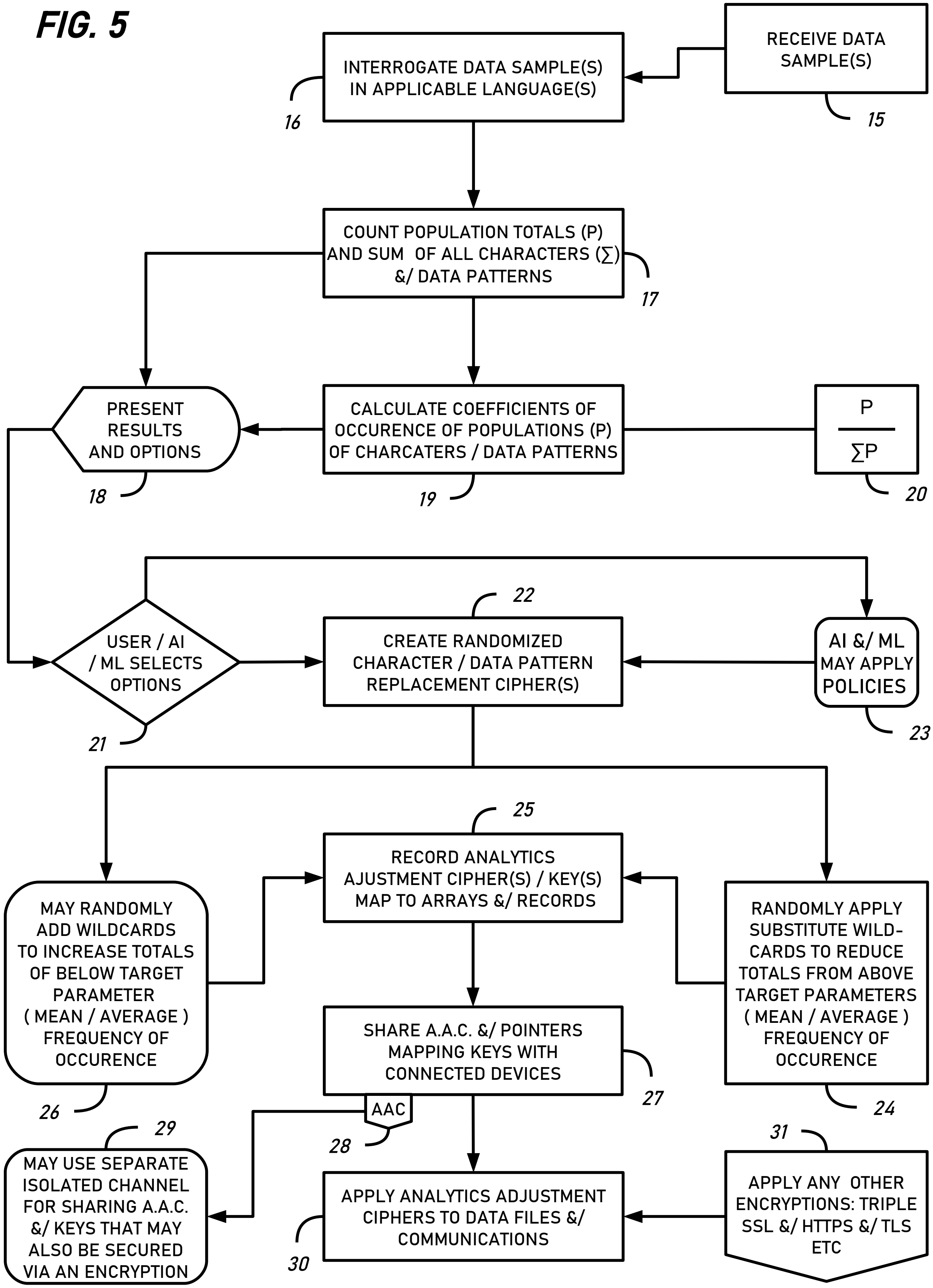

RECEIVE DATA SAMPLE(S)
15
INTERROGATE DATA SAMPLE(S) IN APPLICABLE LANGUAGE(S)
16
COUNT POPULATION TOTALS (P) AND SUM OF ALL CHARACTERS (∑) &/ DATA PATTERNS
17
PRESENT RESULTS AND OPTIONS
18
CALCULATE COEFFICIENTS OF OCCURENCE OF POPULATIONS (P) OF CHARCATERS / DATA PATTERNS
19
P / ∑P
20
USER / AI / ML SELECTS OPTIONS
21
CREATE RANDOMIZED CHARACTER / DATA PATTERN REPLACEMENT CIPHER(S)
22
AI &/ ML MAY APPLY POLICIES
23
RECORD ANALYTICS AJUSTMENT CIPHER(S) / KEY(S) MAP TO ARRAYS &/ RECORDS
25
MAY RANDOMLY ADD WILDCARDS TO INCREASE TOTALS OF BELOW TARGET PARAMETER (MEAN / AVERAGE) FREQUENCY OF OCCURENCE
26
RANDOMLY APPLY SUBSTITUTE WILD-CARDS TO REDUCE TOTALS FROM ABOVE TARGET PARAMETERS (MEAN / AVERAGE) FREQUENCY OF OCCURENCE
24
SHARE A.A.C. &/ POINTERS MAPPING KEYS WITH CONNECTED DEVICES
27
AAC
28
MAY USE SEPARATE ISOLATED CHANNEL FOR SHARING A.A.C. &/ KEYS THAT MAY ALSO BE SECURED VIA AN ENCRYPTION
29
APPLY ANALYTICS ADJUSTMENT CIPHERS TO DATA FILES &/ COMMUNICATIONS
30
APPLY ANY OTHER ENCRYPTIONS: TRIPLE SSL &/ HTTPS &/ TLS ETC
31

FIG. 6

BEFORE ANALYTICS ADJUSTMENT

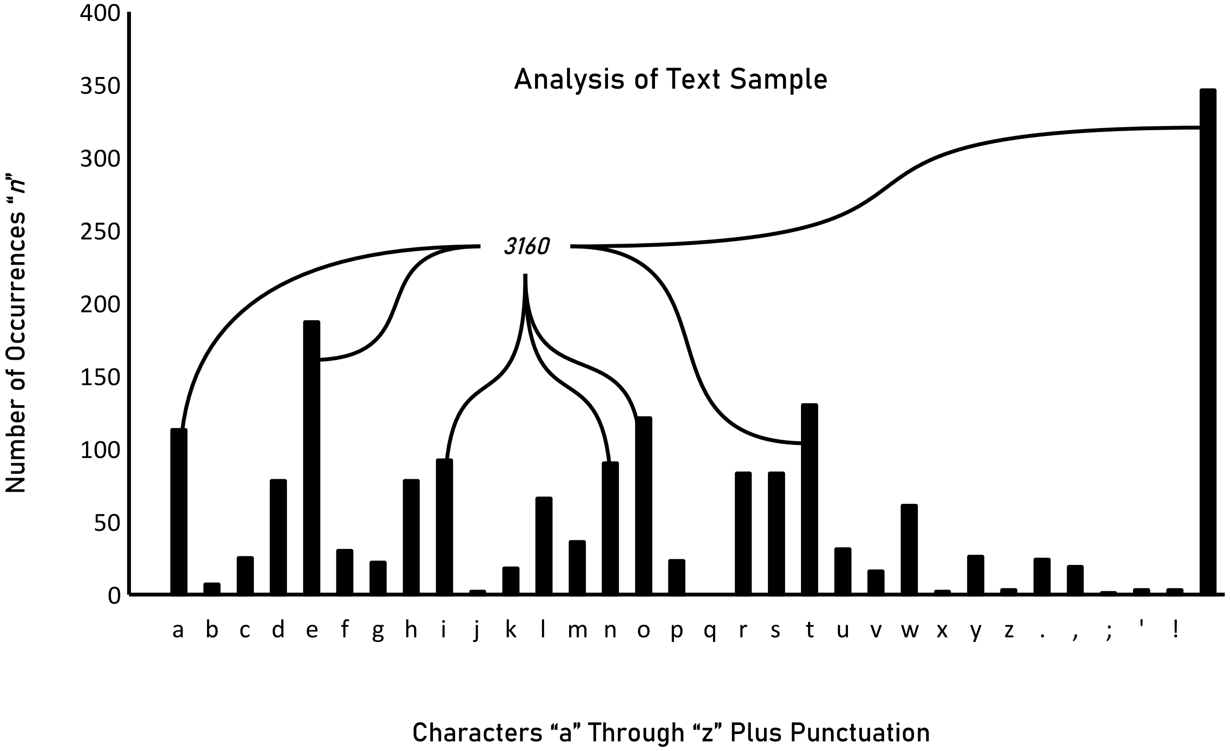


FIG. 6A

AFTER ANALYTICS ADJUSTING CIPHER PROCESS

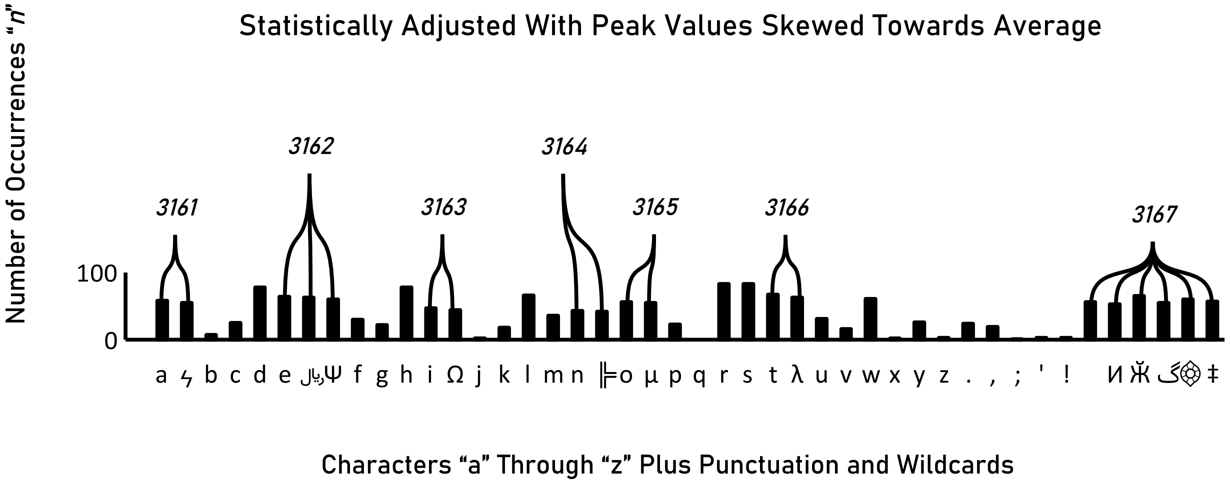


FIG. 7

35 DEFINED PROCESS

CREATE ANALYTICS ADJUSTING CIPHER

36. Receive data sample.

37. Count total identifiable data objects or characters.

38. Identify every type of data object or character.

39. Count totals for every data object or character.

40. Calculate the mean, median, and / or average values for each data object or character.

41. Select adjustment parameters:

42. adjust by random substitutions with wildcards the items with populations of above a parameter down to within that parameter (which may be a percentage of or within a deviation from the mean or median or average population size),

43. (optional) adjust by random additions of (disinformation) wildcards the to items with populations of below a parameter (which may be a percentage of or within a deviation from the mean or median or average population size),

44. (optional) substitute the remaining original data objects and / or characters within a population using wildcards.

45. (optional alternative to step 44) apply another cipher such as Time Randomizing Interface Protocol Language Encryption (TRIPLE) and / or any other compatible cipher and / or mathematical encryption.

46. AAC and other ciphers may be shared with correspondents, and / or negotiated with correspondents via a separate channel that is isolated from the channel that is used to carry the data to be encrypted by the cipher(s).

47. The isolated cipher and / or key channel (if used as such) may be further protected by encryption which may be a strong encryption such as quantum encryption, and / or quantum-anti-interception measures.

48. Optionally, a purely mathematical encryption may be applied to the encrypted data. (Where AAC and / or TRIPLE ciphers are run within a webpage or application that may be running under HTTPS, and / or using other encryption in the state of the art as of 2025, or the future state of the art during the life of this patent.)

FIG. 8

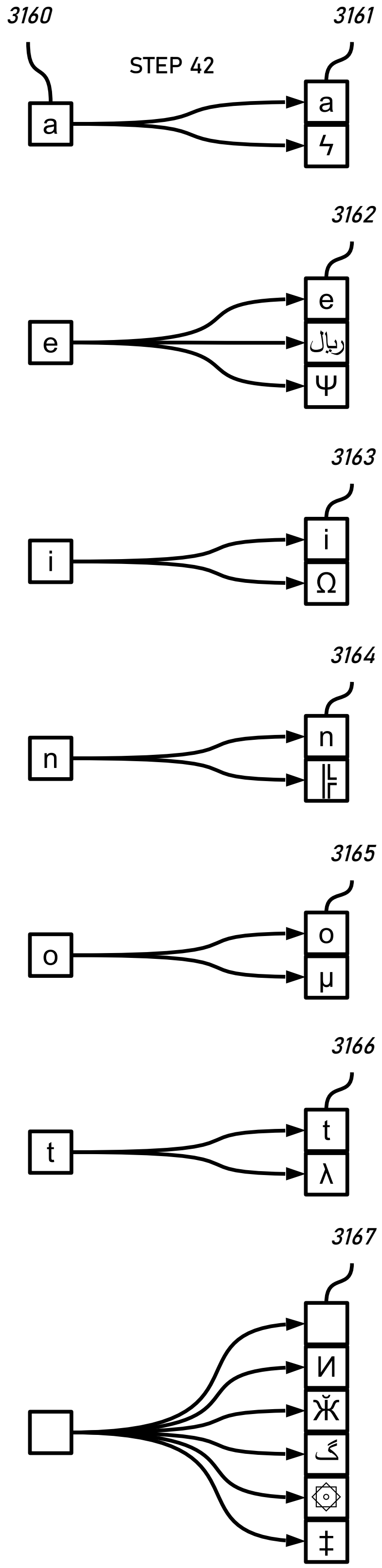

3160
STEP 42
3161
a
a
ϟ
3162
e
e
ريال
Ψ
3163
i
i
Ω
3164
n
n
╠
3165
o
o
μ
3166
t
t
λ
3167
И
Ӂ
گ
۞
‡

FIG. 8A

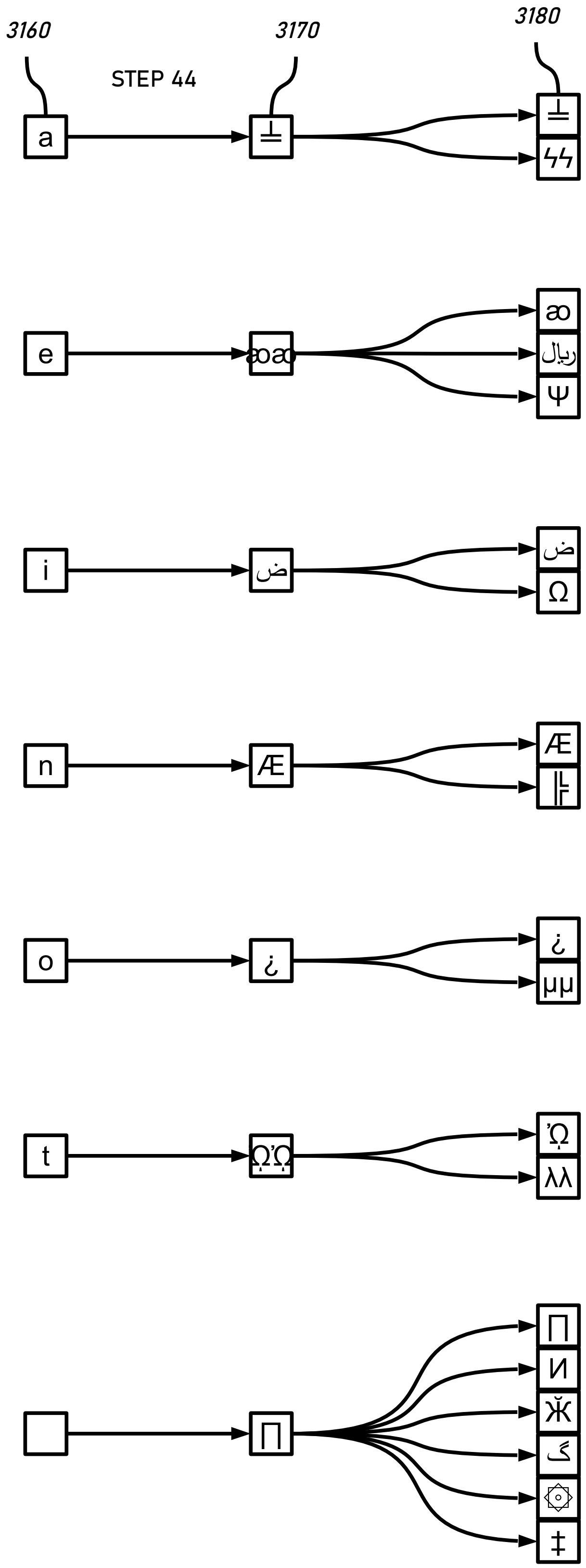

3160
STEP 44
3170
3180
a
╧
╧
ϟϟ
e
ꜵꜵ
ꜵ
ريال
Ψ
i
ض
ض
Ω
n
Æ
Æ
╠
o
¿
¿
μμ
t
Ὠ̣Ὠ̣
Ὠ̣
λλ
∏
∏
И
Ӂ
گ
۞
‡

FIG. 9

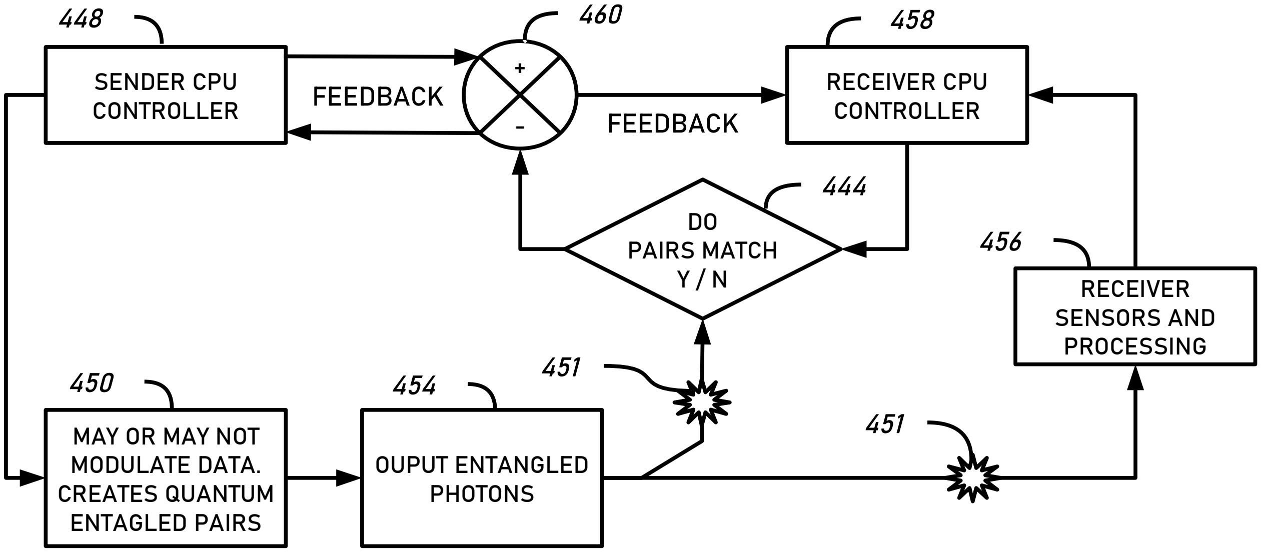

448
SENDER CPU CONTROLLER
FEEDBACK
460
+
-
FEEDBACK
458
RECEIVER CPU CONTROLLER
444
DO PAIRS MATCH Y / N
456
RECEIVER SENSORS AND PROCESSING
451
450
MAY OR MAY NOT MODULATE DATA. CREATES QUANTUM ENTAGLED PAIRS
454
OUPUT ENTANGLED PHOTONS
451

FIG. 9A

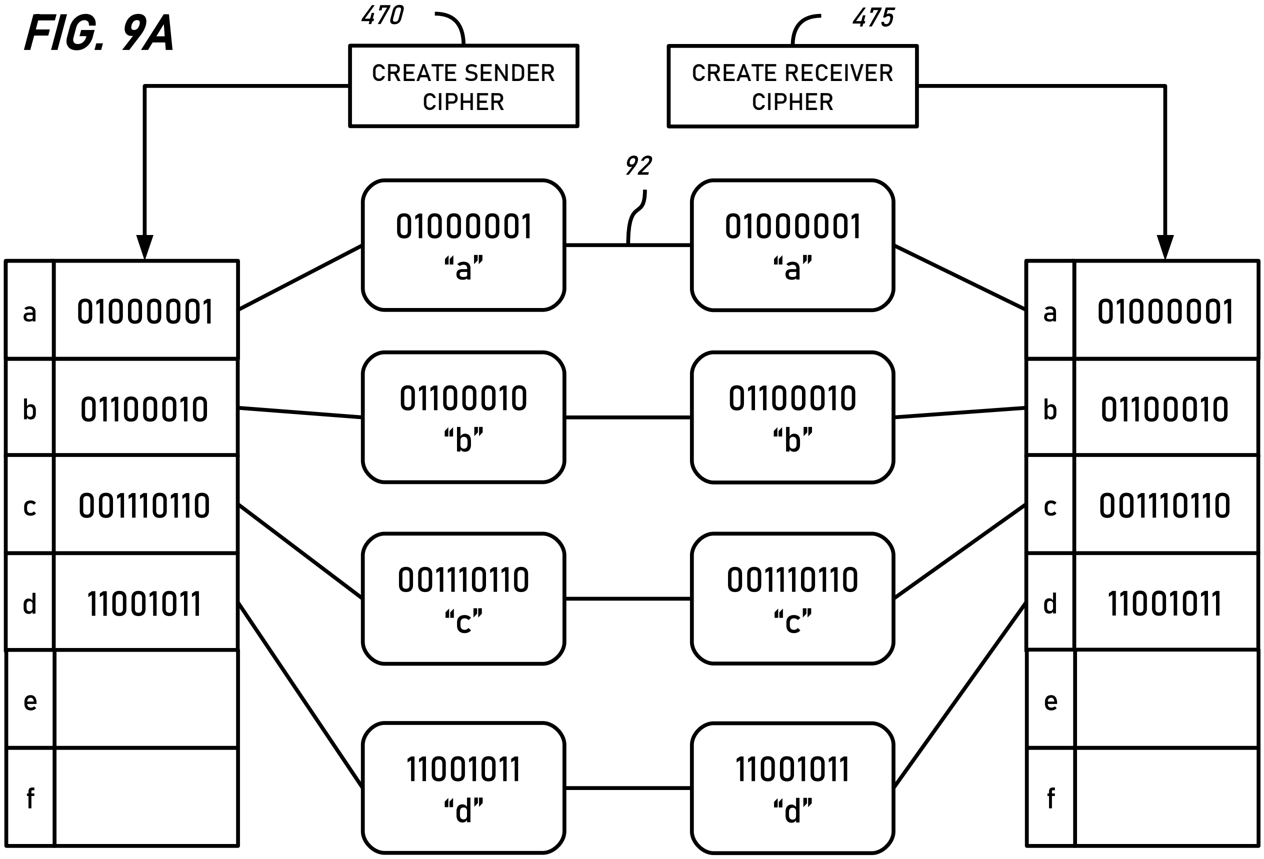
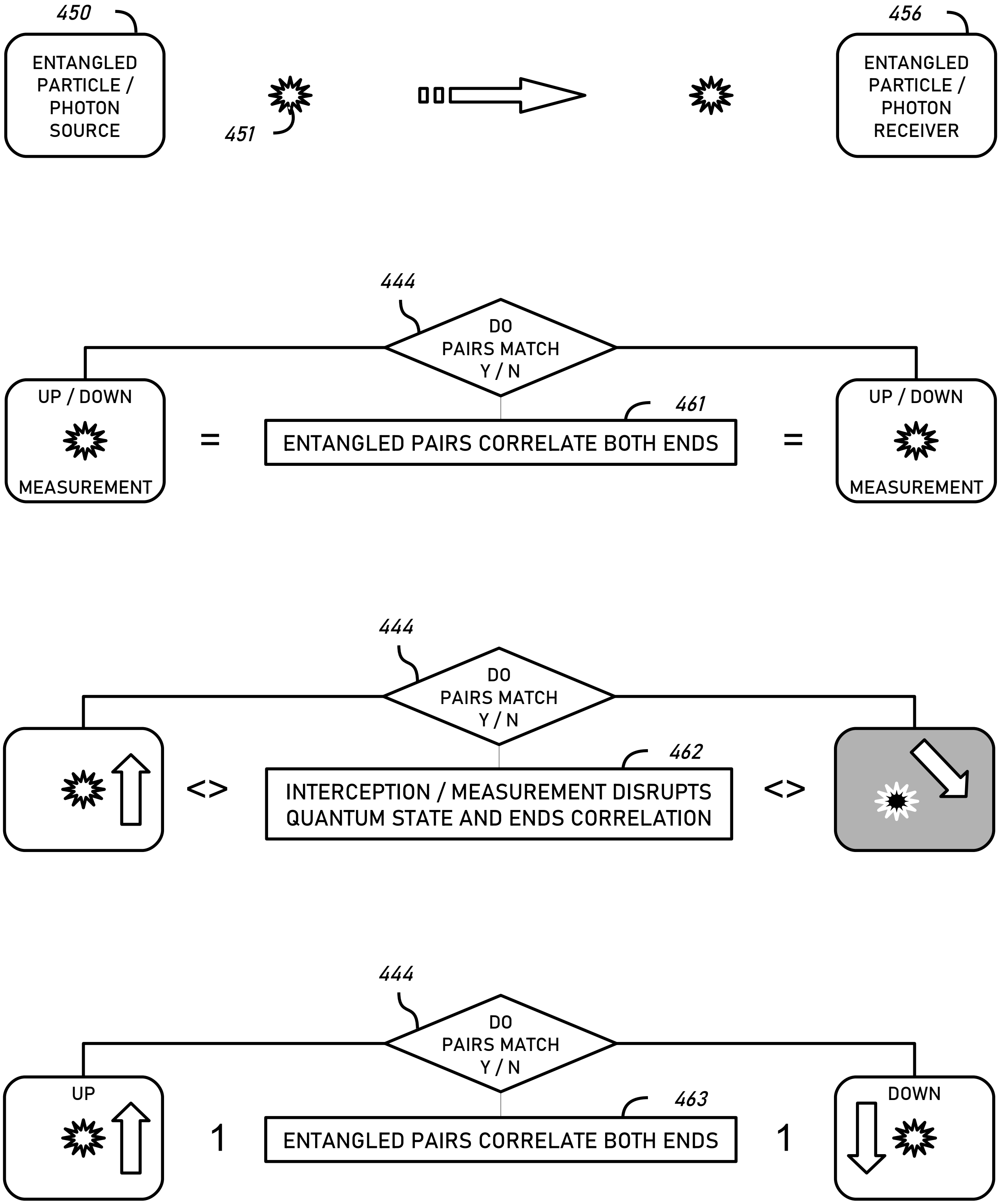

470
CREATE SENDER CIPHER
475
CREATE RECEIVER CIPHER
92
a
01000001
b
01100010
c
001110110
d
11001011
e
f
01000001
"a"
01100010
"b"
001110110
"c"
11001011
"d"
01000001
"a"
01100010
"b"
001110110
"c"
11001011
"d"
a
01000001
b
01100010
c
001110110
d
11001011
e
f

FIG. 10


450
ENTANGLED PARTICLE / PHOTON SOURCE
451
456
ENTANGLED PARTICLE / PHOTON RECEIVER
444
DO PAIRS MATCH Y / N
UP / DOWN
MEASUREMENT
=
461
ENTANGLED PAIRS CORRELATE BOTH ENDS
=
UP / DOWN
MEASUREMENT
444
DO PAIRS MATCH Y / N
<>
462
INTERCEPTION / MEASUREMENT DISRUPTS QUANTUM STATE AND ENDS CORRELATION
<>
444
DO PAIRS MATCH Y / N
UP
1
463
ENTANGLED PAIRS CORRELATE BOTH ENDS
1
DOWN

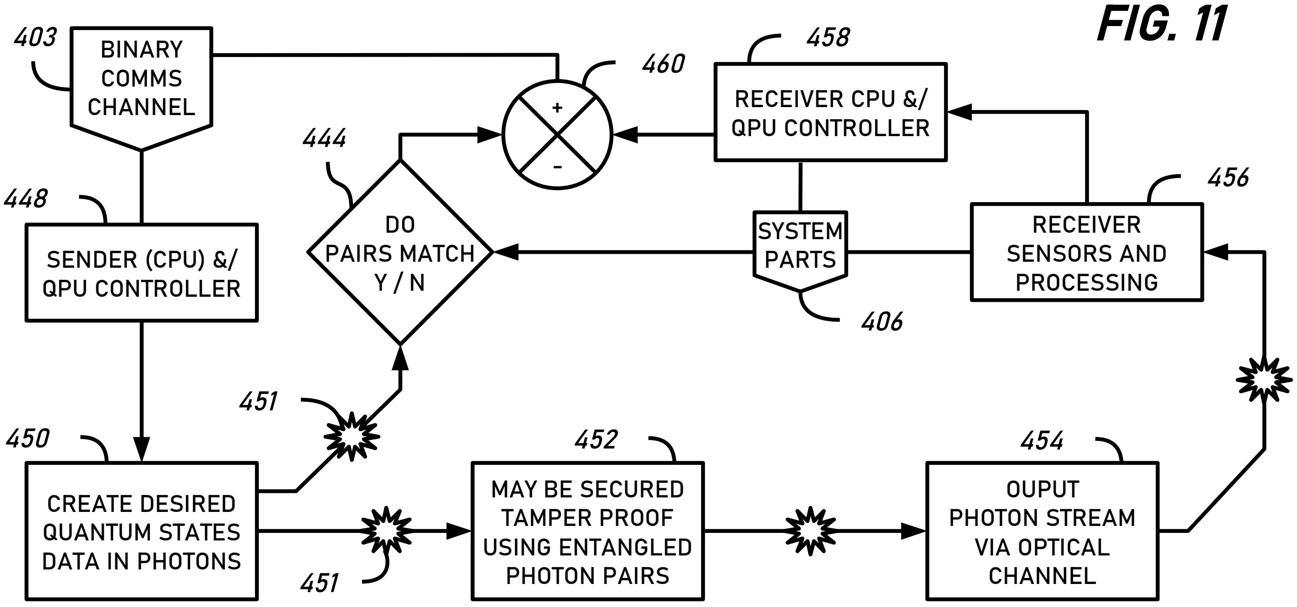

403
BINARY COMMS CHANNEL
460
+
-
458
RECEIVER CPU &/ QPU CONTROLLER
444
DO PAIRS MATCH Y / N
448
SENDER (CPU) &/ QPU CONTROLLER
SYSTEM PARTS
406
456
RECEIVER SENSORS AND PROCESSING
451
450
CREATE DESIRED QUANTUM STATES DATA IN PHOTONS
451
452
MAY BE SECURED TAMPER PROOF USING ENTANGLED PHOTON PAIRS
454
OUPUT PHOTON STREAM VIA OPTICAL CHANNEL

FIG. 11

FIG. 11A

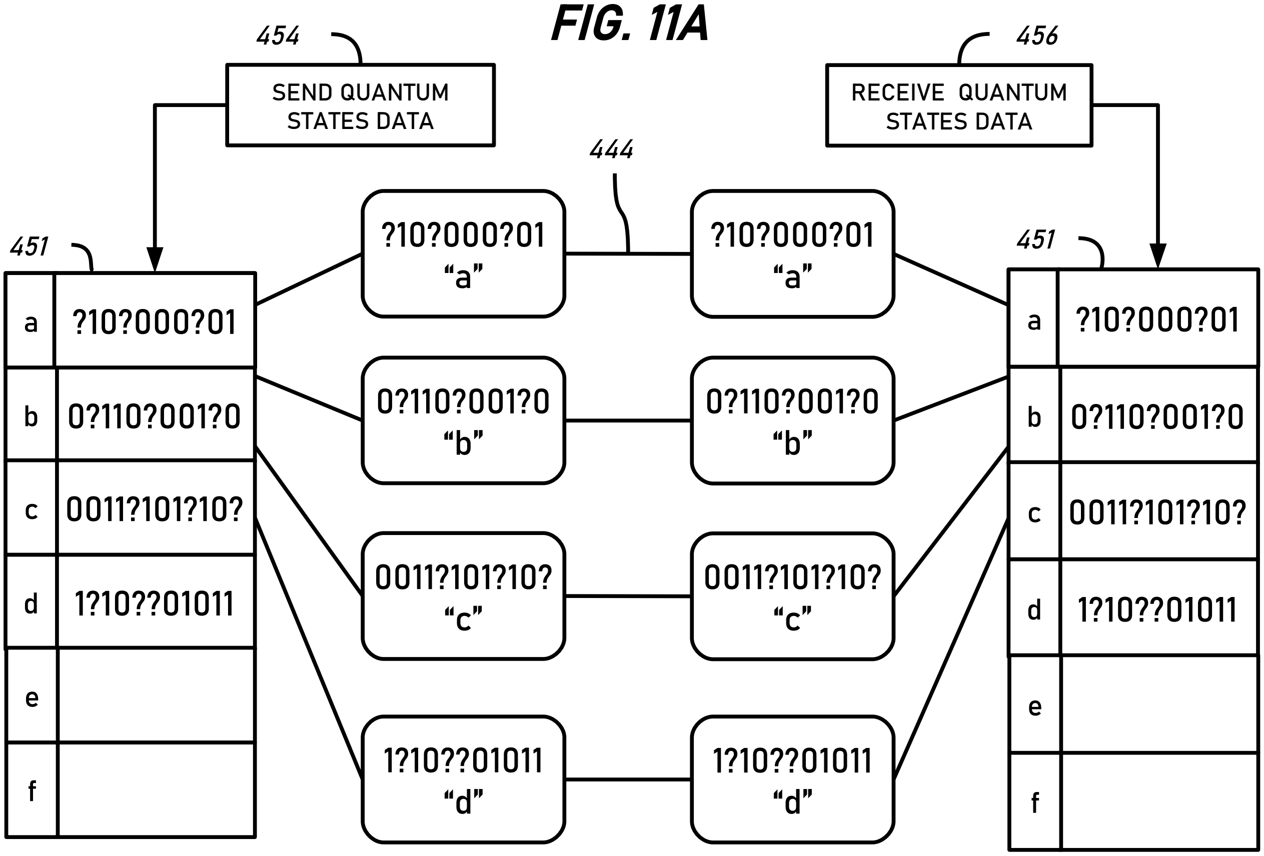


FIG. 12

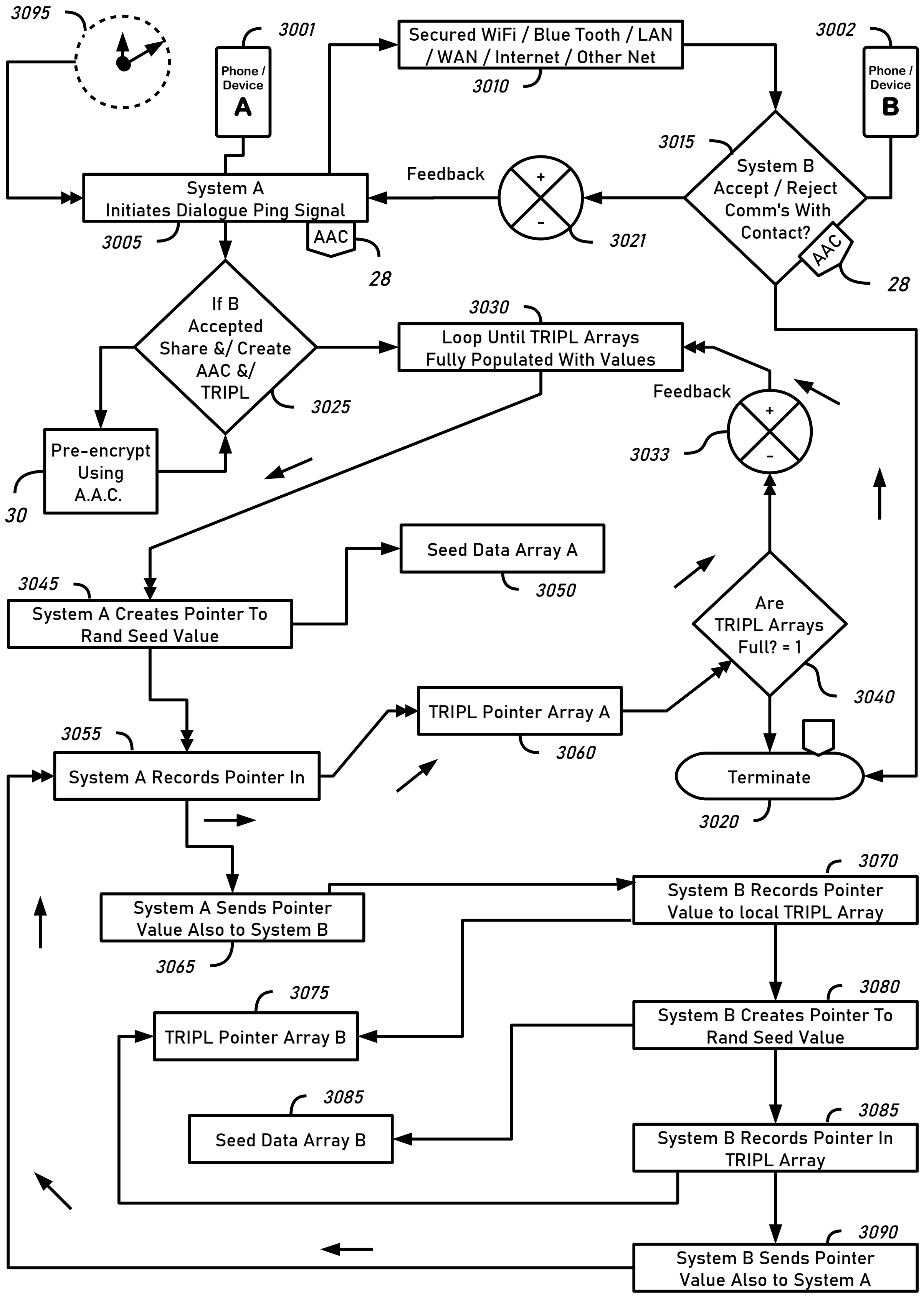
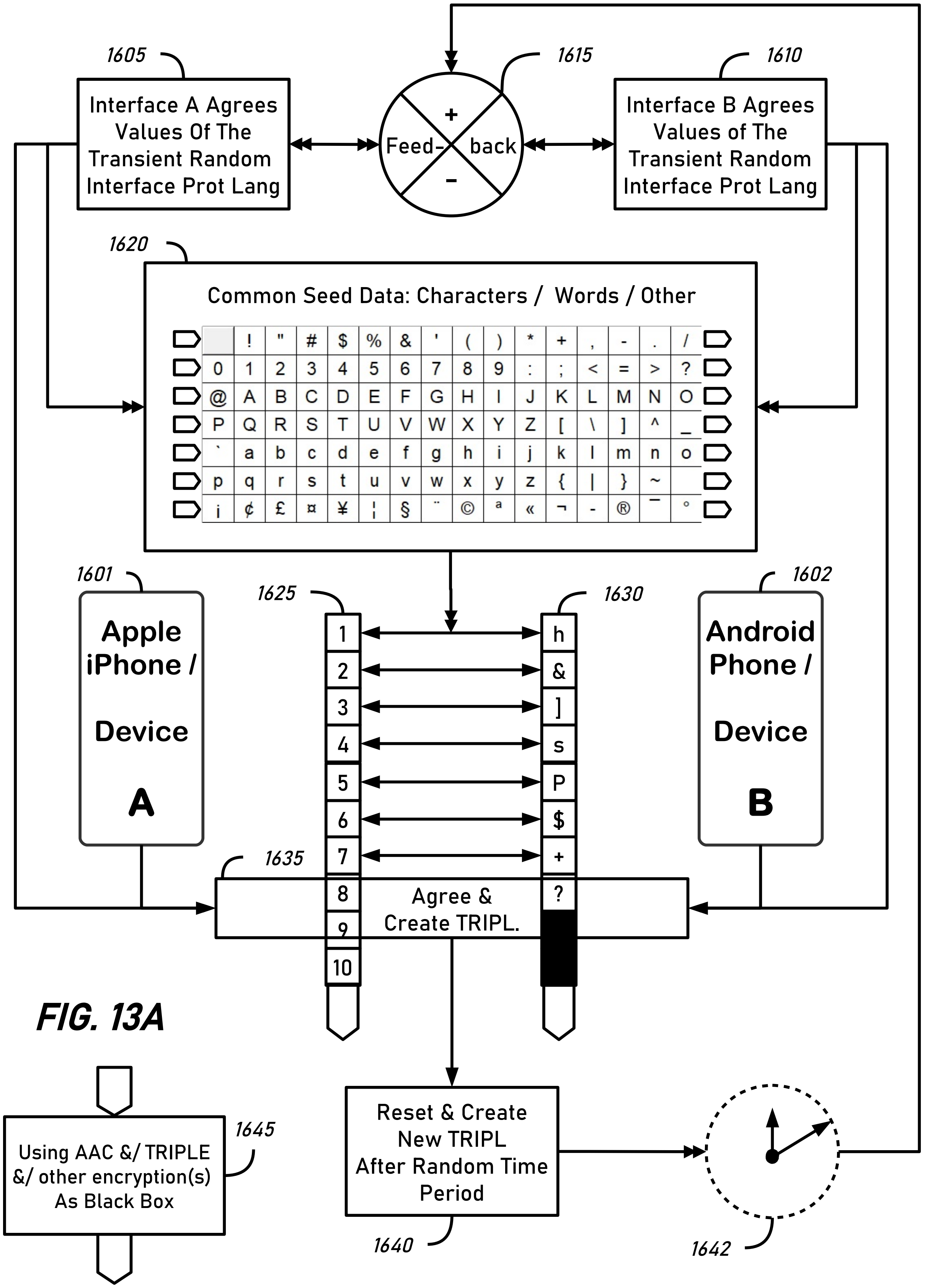

3095
3001
Phone / Device A
Secured WiFi / Blue Tooth / LAN / WAN / Internet / Other Net
3010
3002
Phone / Device B
3015
System B Accept / Reject Comm's With Contact?
AAC
28
Feedback
+
-
3021
System A Initiates Dialogue Ping Signal
3005
AAC
28
If B Accepted Share &/ Create AAC &/ TRIPL
3025
3030
Loop Until TRIPL Arrays Fully Populated With Values
Feedback
+
-
3033
Pre-encrypt Using A.A.C.
30
Seed Data Array A
3050
3045
System A Creates Pointer To Rand Seed Value
Are TRIPL Arrays Full? = 1
3040
3055
System A Records Pointer In
TRIPL Pointer Array A
3060
Terminate
3020
3070
System A Sends Pointer Value Also to System B
System B Records Pointer Value to local TRIPL Array
3065
3075
3080
TRIPL Pointer Array B
System B Creates Pointer To Rand Seed Value
3085
Seed Data Array B
3085
System B Records Pointer In TRIPL Array
3090
System B Sends Pointer Value Also to System A

FIG. 13


1605
Interface A Agrees Values Of The Transient Random Interface Prot Lang
1615
+
Feed-
back
–
1610
Interface B Agrees Values of The Transient Random Interface Prot Lang
1620
Common Seed Data: Characters / Words / Other
1601
Apple iPhone / Device A
1625
1630
1602
Android Phone / Device B
1 h
2 &
3]
4 s
5 P
6 $
7 +
8 ?
9
10
1635
Agree & Create TRIPL.
Reset & Create New TRIPL After Random Time Period
1640
1642
Using AAC &/ TRIPLE &/ other encryption(s) As Black Box
1645

	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯	°

FIG. 13A

FIG. 14

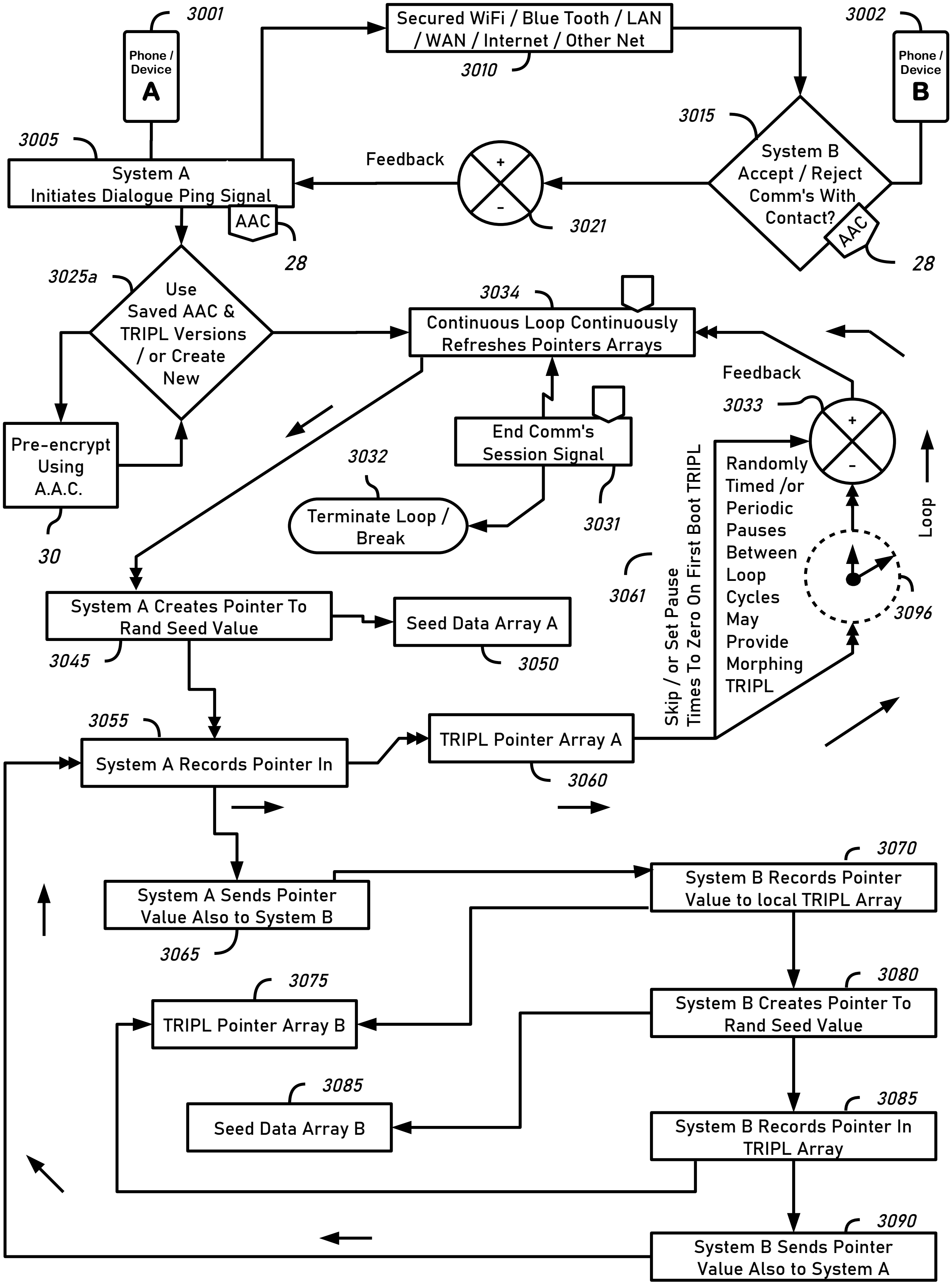

3001
Phone / Device A
Secured WiFi / Blue Tooth / LAN / WAN / Internet / Other Net
3010
3002
Phone / Device B
3015
3005
System A Initiates Dialogue Ping Signal
Feedback
+
-
3021
System B Accept / Reject Comm's With Contact?
AAC
28
AAC
28
3025a
Use Saved AAC & TRIPL Versions / or Create New
3034
Continuous Loop Continuously Refreshes Pointers Arrays
Feedback
3033
+
-
End Comm's Session Signal
3032
3031
Terminate Loop / Break
Pre-encrypt Using A.A.C.
30
Skip / or Set Pause Times To Zero On First Boot TRIPL
Randomly Timed /or Periodic Pauses Between Loop Cycles May Provide Morphing TRIPL
Loop
3061
3096
System A Creates Pointer To Rand Seed Value
Seed Data Array A
3045
3050
3055
System A Records Pointer In
TRIPL Pointer Array A
3060
System A Sends Pointer Value Also to System B
3065
3070
System B Records Pointer Value to local TRIPL Array
3075
TRIPL Pointer Array B
3080
System B Creates Pointer To Rand Seed Value
3085
Seed Data Array B
3085
System B Records Pointer In TRIPL Array
3090
System B Sends Pointer Value Also to System A

FIG. 15

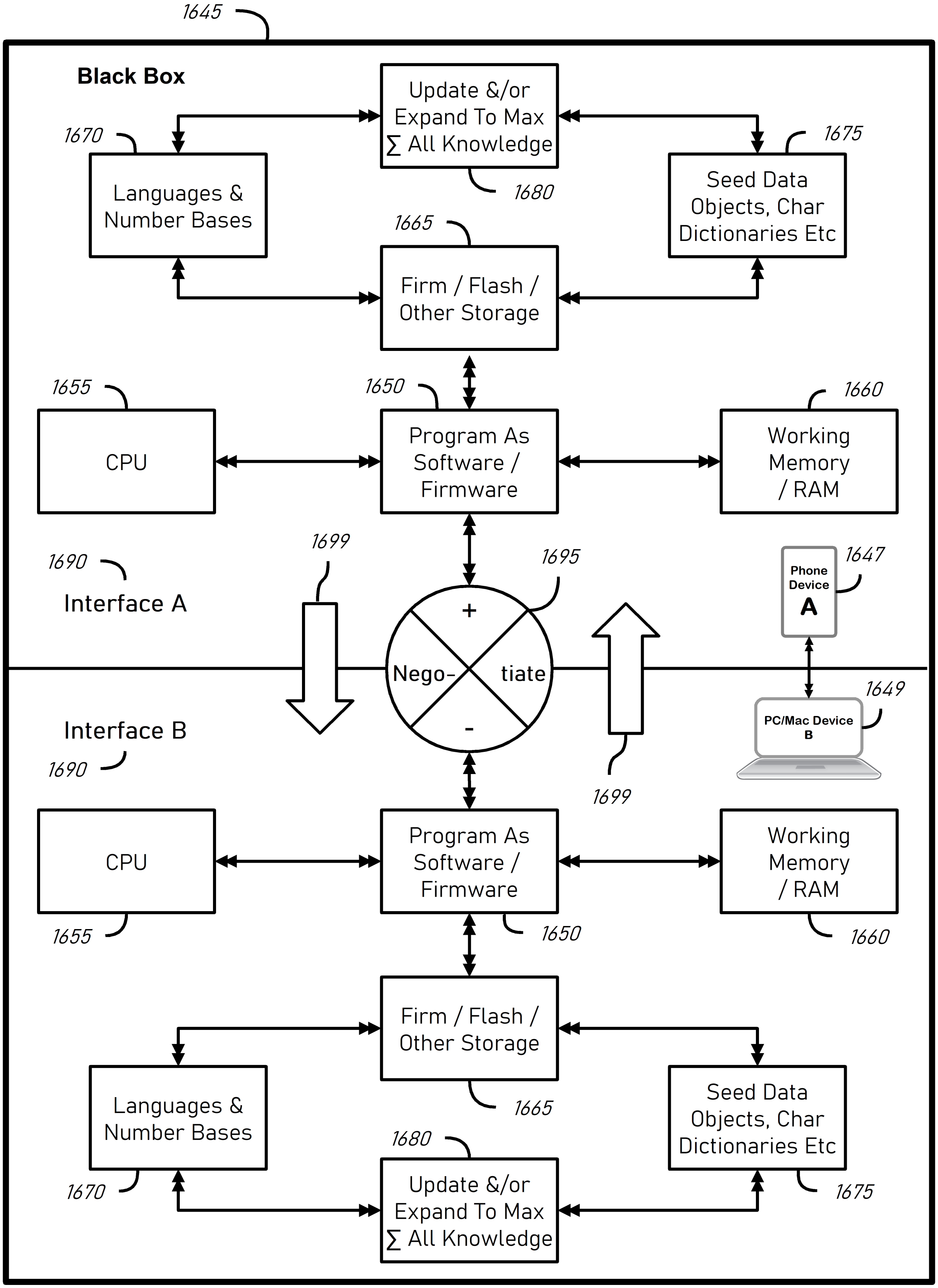
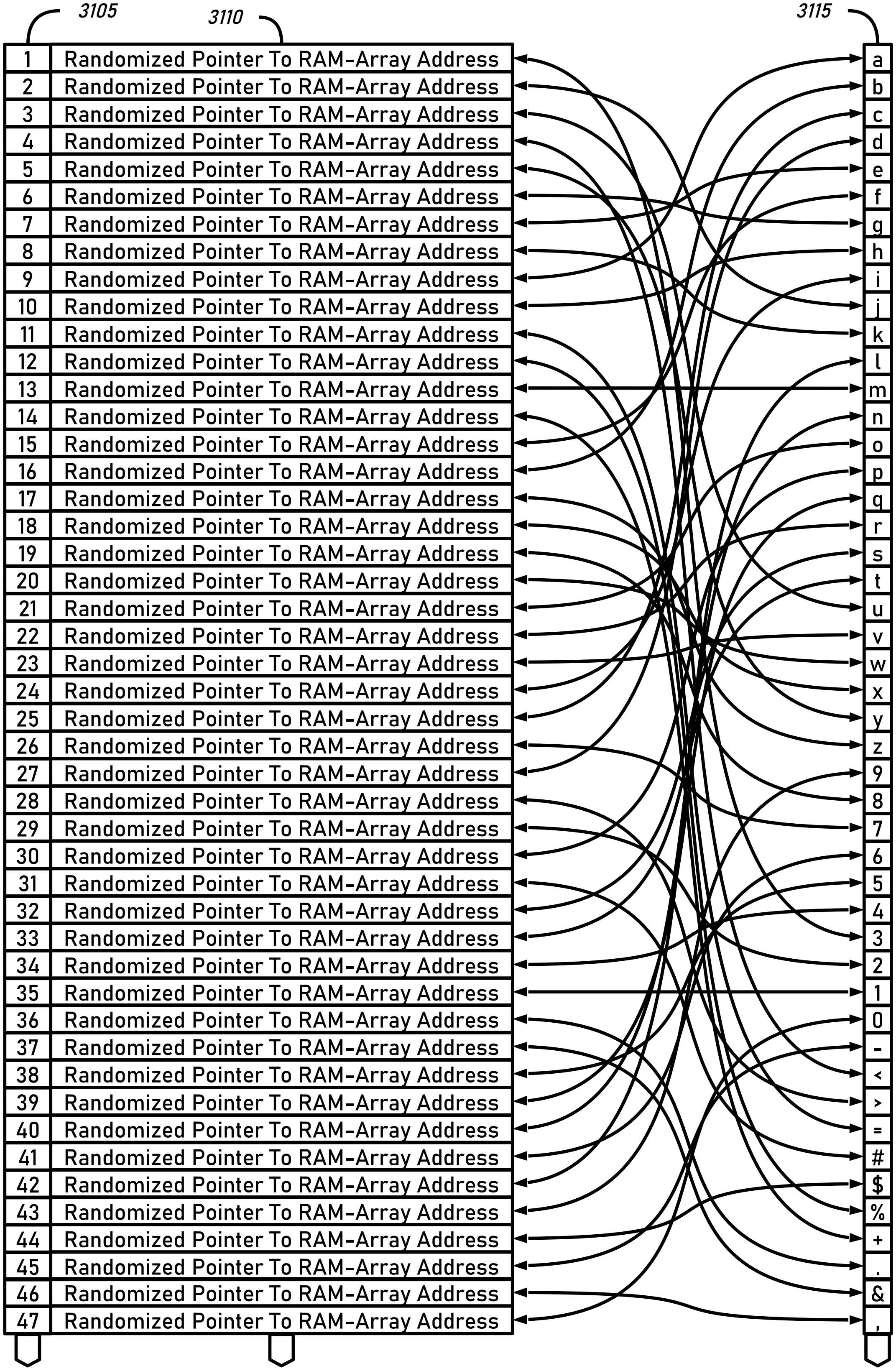

1645
Black Box
Update &/or Expand To Max ∑ All Knowledge
1680
1670
Languages & Number Bases
1675
Seed Data Objects, Char Dictionaries Etc
1665
Firm / Flash / Other Storage
1655
CPU
1650
Program As Software / Firmware
1660
Working Memory / RAM
1690
Interface A
1699
1695
+
Nego-
tiate
-
1647
Phone Device A
1649
PC/Mac Device B
Interface B
1690
1699
CPU
1655
Program As Software / Firmware
1650
Working Memory / RAM
1660
Firm / Flash / Other Storage
1665
Languages & Number Bases
1670
Seed Data Objects, Char Dictionaries Etc
1675
1680
Update &/or Expand To Max ∑ All Knowledge

FIG. 16


3105
3110
3115
1 Randomized Pointer To RAM-Array Address
2 Randomized Pointer To RAM-Array Address
3 Randomized Pointer To RAM-Array Address
4 Randomized Pointer To RAM-Array Address
5 Randomized Pointer To RAM-Array Address
6 Randomized Pointer To RAM-Array Address
7 Randomized Pointer To RAM-Array Address
8 Randomized Pointer To RAM-Array Address
9 Randomized Pointer To RAM-Array Address
10 Randomized Pointer To RAM-Array Address
11 Randomized Pointer To RAM-Array Address
12 Randomized Pointer To RAM-Array Address
13 Randomized Pointer To RAM-Array Address
14 Randomized Pointer To RAM-Array Address
15 Randomized Pointer To RAM-Array Address
16 Randomized Pointer To RAM-Array Address
17 Randomized Pointer To RAM-Array Address
18 Randomized Pointer To RAM-Array Address
19 Randomized Pointer To RAM-Array Address
20 Randomized Pointer To RAM-Array Address
21 Randomized Pointer To RAM-Array Address
22 Randomized Pointer To RAM-Array Address
23 Randomized Pointer To RAM-Array Address
24 Randomized Pointer To RAM-Array Address
25 Randomized Pointer To RAM-Array Address
26 Randomized Pointer To RAM-Array Address
27 Randomized Pointer To RAM-Array Address
28 Randomized Pointer To RAM-Array Address
29 Randomized Pointer To RAM-Array Address
30 Randomized Pointer To RAM-Array Address
31 Randomized Pointer To RAM-Array Address
32 Randomized Pointer To RAM-Array Address
33 Randomized Pointer To RAM-Array Address
34 Randomized Pointer To RAM-Array Address
35 Randomized Pointer To RAM-Array Address
36 Randomized Pointer To RAM-Array Address
37 Randomized Pointer To RAM-Array Address
38 Randomized Pointer To RAM-Array Address
39 Randomized Pointer To RAM-Array Address
40 Randomized Pointer To RAM-Array Address
41 Randomized Pointer To RAM-Array Address
42 Randomized Pointer To RAM-Array Address
43 Randomized Pointer To RAM-Array Address
44 Randomized Pointer To RAM-Array Address
45 Randomized Pointer To RAM-Array Address
46 Randomized Pointer To RAM-Array Address
47 Randomized Pointer To RAM-Array Address
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
x
y
z
9
8
7
6
5
4
3
2
1
0
-
<
>
=
#
$
%
+
.
&
,

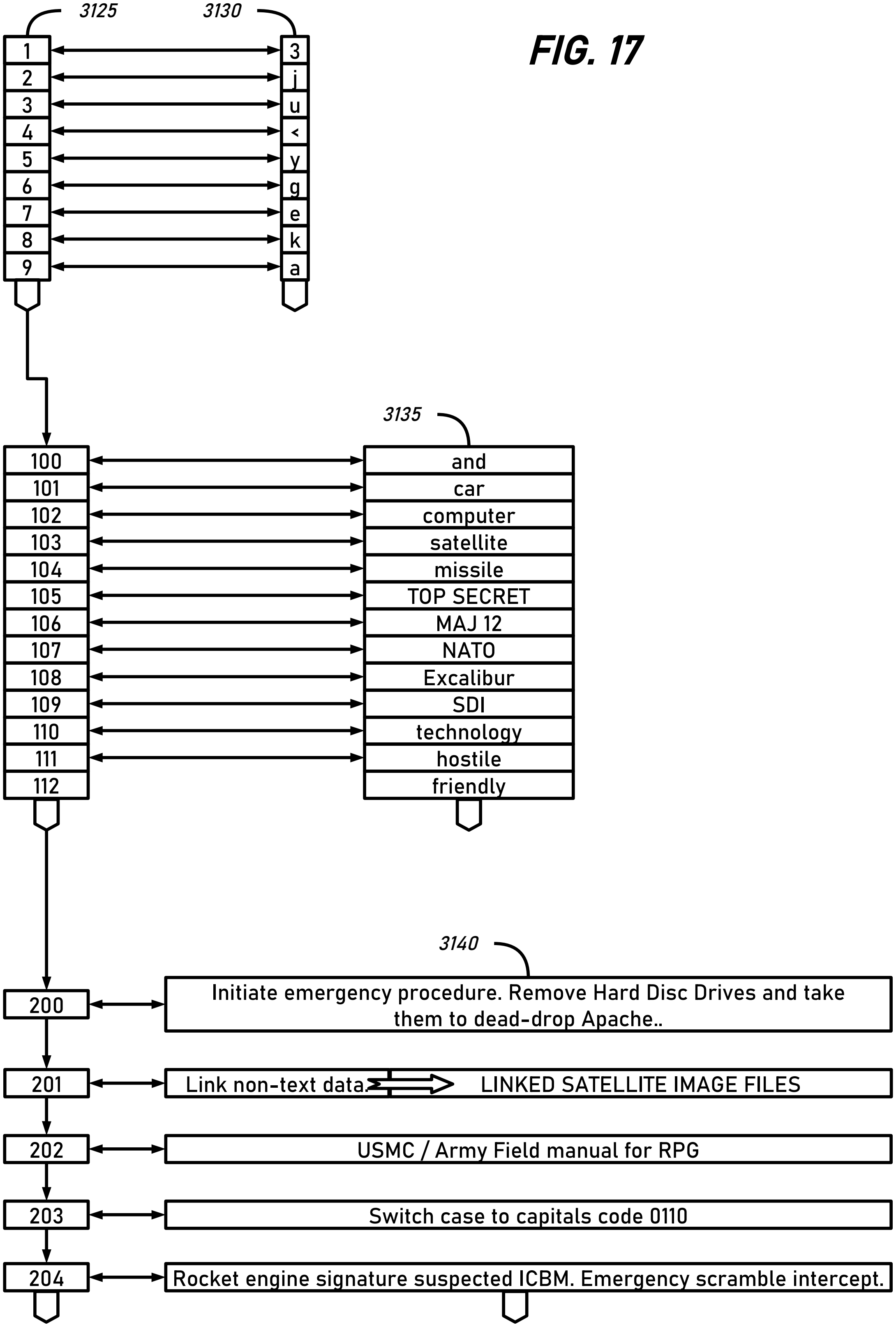

3125
3130
1
2
3
4
5
6
7
8
9
3
j
u
<
y
g
e
k
a
3135
100
101
102
103
104
105
106
107
108
109
110
111
112
and
car
computer
satellite
missile
TOP SECRET
MAJ 12
NATO
Excalibur
SDI
technology
hostile
friendly
3140
200
Initiate emergency procedure. Remove Hard Disc Drives and take them to dead-drop Apache..
201
Link non-text data.
LINKED SATELLITE IMAGE FILES
202
USMC / Army Field manual for RPG
203
Switch case to capitals code 0110
204
Rocket engine signature suspected ICBM. Emergency scramble intercept.

FIG. 17

FIG. 18

3150

DEFINED PROCESS

RUN-TIME EXAMPLE OF POSSIBLE TRIPL CREATION PROCESS

Device A initiates TRIPL creation via request to Device B

Device B agrees

Device A generates a pointer to its seed data array which points to “3” the pointer / value is sent to device B

Both devices record the pointer to “3” at their 1st TRIPL array index location, or with pointer index value 1

Device B generates a pointer to its seed data array which points to “j” the pointer / value is sent to device A

Both devices record the pointer to “j” at their 2nd TRIPL array index location, or with pointer index value 2

Device A generates a pointer to its seed data array which points to “u” the pointer / value is sent to device B

Both devices record the pointer to “u” at their 3rd TRIPL array index location, or with pointer index value 3

Device B generates a pointer to its seed data array which points to “<” the pointer / value is sent to device A

Both devices record the pointer to “<” at their 4th TRIPL array index location, or with pointer index value 4

Device A generates a pointer to its seed data array which points to “y” the pointer / value is sent to device B

Both devices record the pointer to “y” at their 5th TRIPL array index location, or with pointer index value 5

Device B generates a pointer to its seed data array which points to “g” the pointer / value is sent to device A

Both devices record the pointer to “g” at their 6th TRIPL array index location, or with pointer index value 6

-------------- BREAK --------------

Process continues until complete TRIPL is negotiated... An array of characters, character strings, or records with an index capability may be used to store the new TRIPL or an array of pointers to the TRIPL may be used in TRIPL encrypted communications.

TO FIG.18A

FIG. 18A

3150

DEFINED PROCESS

FROM FIG.18

RUN-TIME EXAMPLE OF POSSIBLE TRIPL CREATION PROCESS

Continuation string data objects...

Device A generates a pointer to its seed data array which points to “and” the pointer / value is sent to device B

Both devices record the pointer to “and” at their 100th TRIPL array index location, or with pointer index value 100

Device B generates a pointer to its seed data array which points to “car” the pointer / value is sent to device A

Both devices record the pointer to “car” at their 101st TRIPL array index location, or with pointer index value 101

Device A generates a pointer to its seed data array which points to “computer” the pointer / value is sent to device B

Both devices record the pointer to “computer” at their 102nd TRIPL array index location, or with pointer index value 102

-------------- BREAK --------------

Continuation complex data-objects...

Device B generates a pointer to its seed data array which points to “Initiate emergency procedure. Remove Hard Disc Drives and take them to dead-drop Apache.” the pointer / value is sent to device A

Both devices record the pointer to “Initiate emergency procedure. Remove Hard Disc Drives and take them to dead-drop Apache.” at their 200th TRIPL array index location, or with pointer index value 200

Device A generates a pointer to its seed data array which points to “Link non-text data” the pointer / value is sent to device B

Both devices record the pointer to “Link non-text data” at their 201st TRIPL array index location, or with pointer index value 201

Device B generates a pointer to its seed data array which points to “USMC Field manual” the pointer / value is sent to device A

Both devices record the pointer to “USMC / Army Field manual for RPG” at their 202nd TRIPL array index location, or with pointer index value 202

Process continues until complete TRIPL is negotiated... An array of characters, character strings, or records with an index capability may be used to store the new TRIPL or an array of pointers to the TRIPL may be used in TRIPL encrypted communications.

FIG. 19

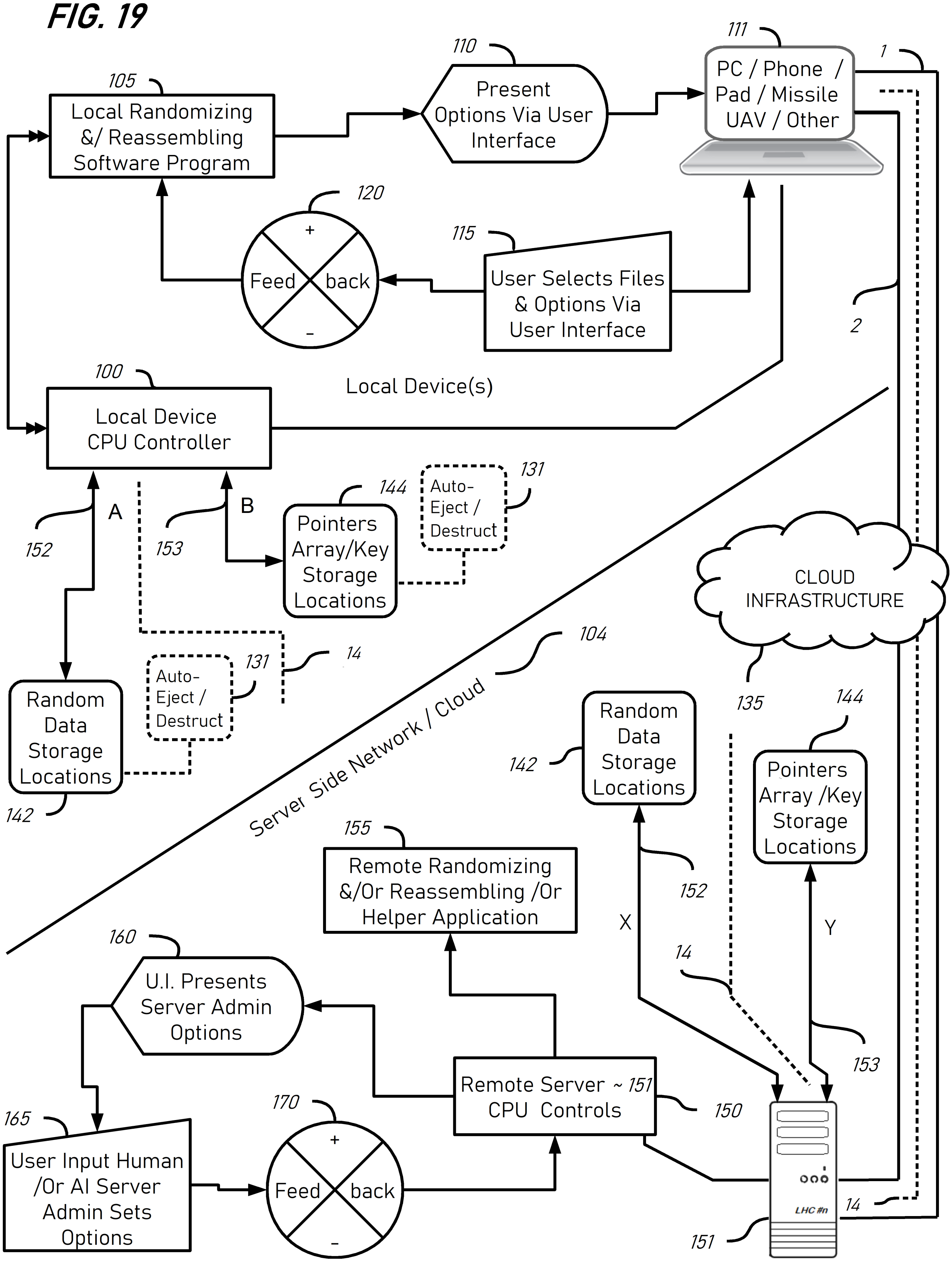

105
Local Randomizing &/ Reassembling Software Program
110
Present Options Via User Interface
111
PC / Phone / Pad / Missile UAV / Other
1
120
+
Feed
back
-
115
User Selects Files & Options Via User Interface
2
100
Local Device CPU Controller
Local Device(s)
131
Auto-Eject / Destruct
144
Pointers Array/Key Storage Locations
A
B
152
153
CLOUD INFRASTRUCTURE
104
14
131
Auto-Eject / Destruct
Random Data Storage Locations
142
Server Side Network / Cloud
135
144
142
Random Data Storage Locations
Pointers Array /Key Storage Locations
155
Remote Randomizing &/Or Reassembling /Or Helper Application
152
X
Y
14
153
160
U.I. Presents Server Admin Options
165
User Input Human /Or AI Server Admin Sets Options
170
+
Feed
back
-
Remote Server ~ 151 CPU Controls
150
LHC #n
14
151

FIG. 19A

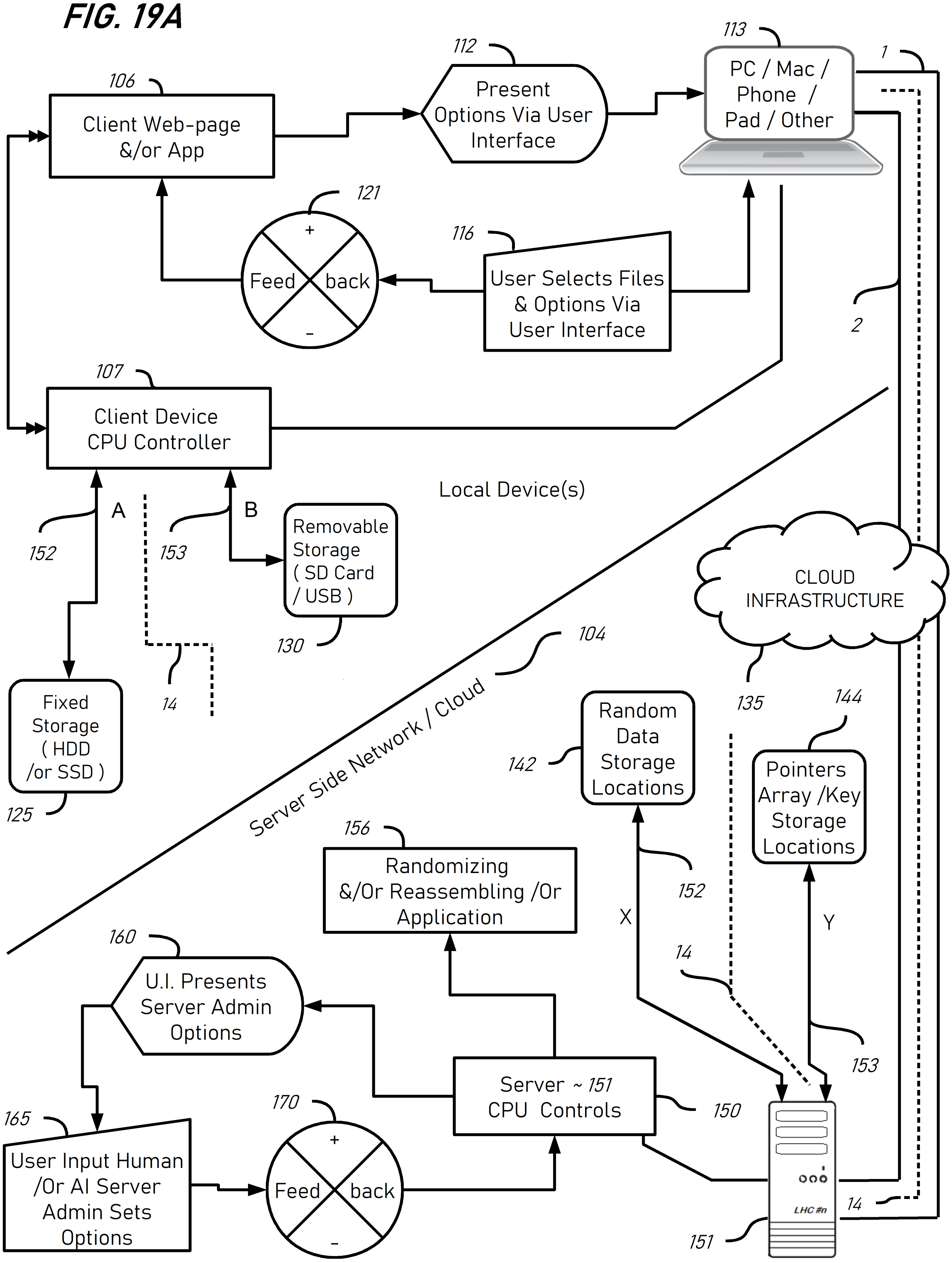

106
Client Web-page &/or App
112
Present Options Via User Interface
113
PC / Mac / Phone / Pad / Other
1
121
+
Feed
back
-
116
User Selects Files & Options Via User Interface
2
107
Client Device CPU Controller
Local Device(s)
A
B
152
153
Removable Storage (SD Card / USB)
130
Fixed Storage (HDD /or SSD)
125
14
CLOUD INFRASTRUCTURE
104
135
Server Side Network / Cloud
Random Data Storage Locations
142
144
Pointers Array /Key Storage Locations
156
Randomizing &/Or Reassembling /Or Application
152
X
Y
14
153
160
U.I. Presents Server Admin Options
Server ~ 151
CPU Controls
150
170
+
Feed
back
-
165
User Input Human /Or AI Server Admin Sets Options
LHC #n
14
151

FIG. 19B

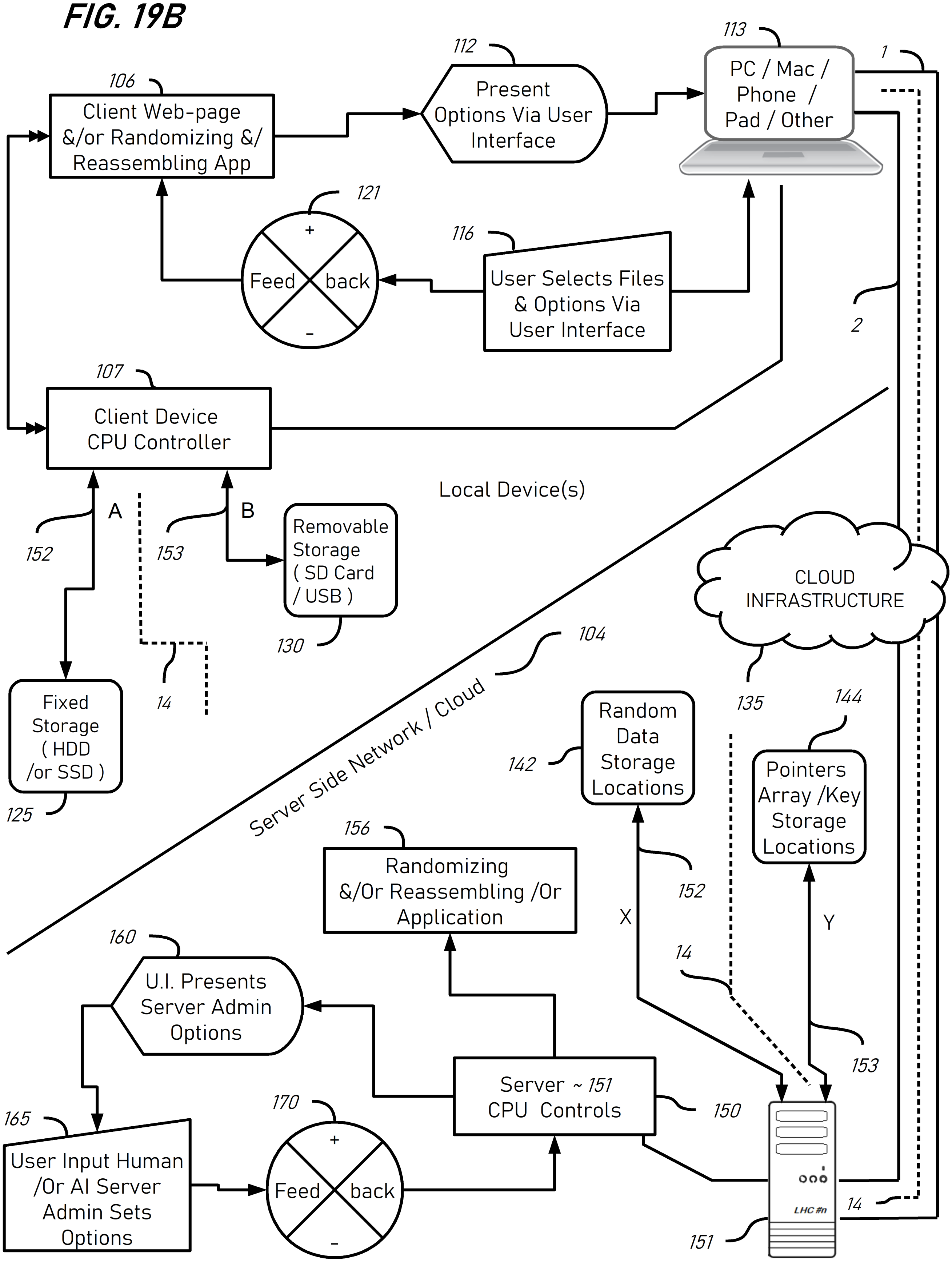

106
Client Web-page &/or Randomizing &/ Reassembling App
112
Present Options Via User Interface
113
PC / Mac / Phone / Pad / Other
1
121
+
Feed
back
-
116
User Selects Files & Options Via User Interface
2
107
Client Device CPU Controller
Local Device(s)
A
B
152
153
Removable Storage (SD Card / USB)
130
CLOUD INFRASTRUCTURE
104
Fixed Storage (HDD /or SSD)
125
14
Server Side Network / Cloud
135
144
142
Random Data Storage Locations
Pointers Array /Key Storage Locations
156
Randomizing &/Or Reassembling /Or Application
152
X
Y
14
160
U.I. Presents Server Admin Options
153
170
+
Feed
back
-
Server ~ 151 CPU Controls
150
165
User Input Human /Or AI Server Admin Sets Options
LHC #n
14
151

FIG. 20

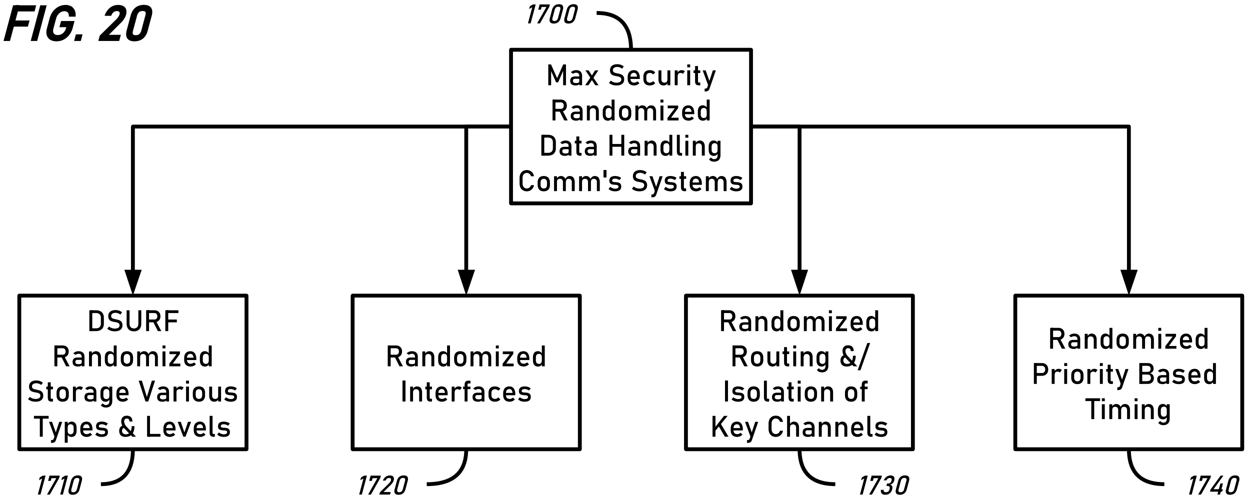

1700
Max Security Randomized Data Handling Comm's Systems
DSURF Randomized Storage Various Types & Levels
1710
Randomized Interfaces
1720
Randomized Routing &/ Isolation of Key Channels
1730
Randomized Priority Based Timing
1740

FIG. 21

PRIOR ART

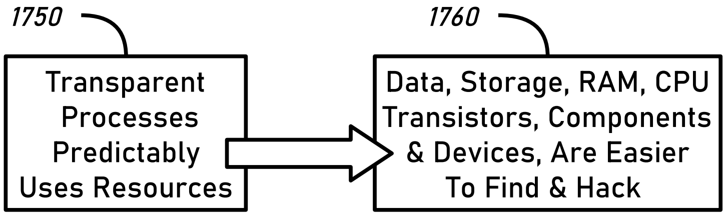

1750
Transparent Processes Predictably Uses Resources
1760
Data, Storage, RAM, CPU Transistors, Components & Devices, Are Easier To Find & Hack

FIG. 22

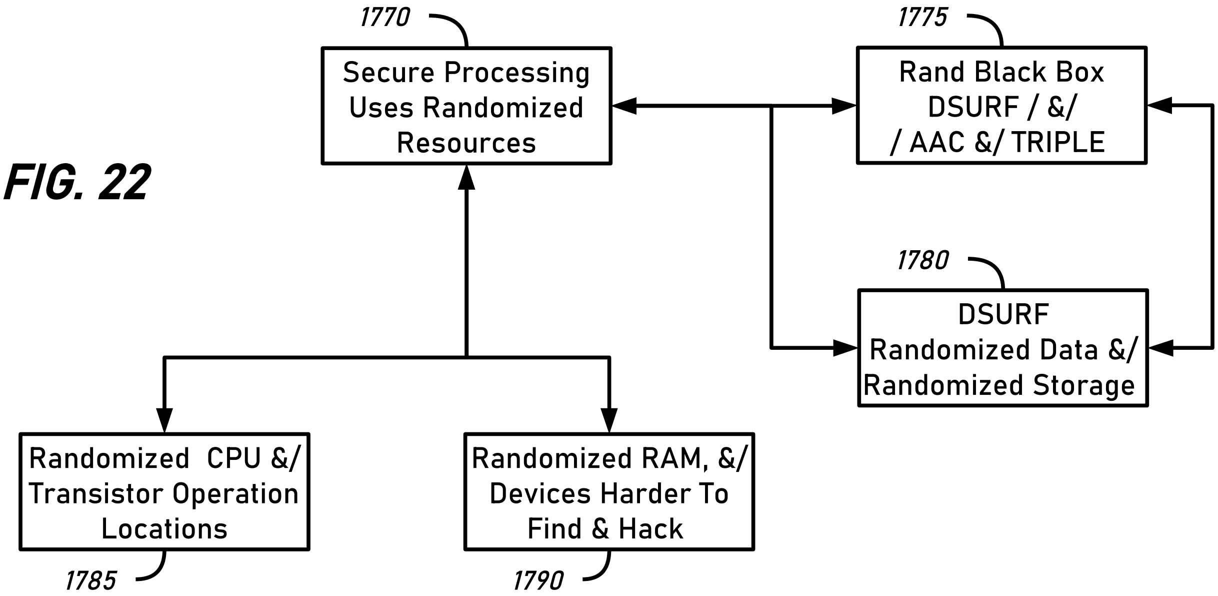

1770
Secure Processing Uses Randomized Resources
1775
Rand Black Box DSURF / &/ / AAC &/ TRIPLE
1780
DSURF Randomized Data &/ Randomized Storage
Randomized CPU &/ Transistor Operation Locations
1785
Randomized RAM, &/ Devices Harder To Find & Hack
1790

FIG. 23

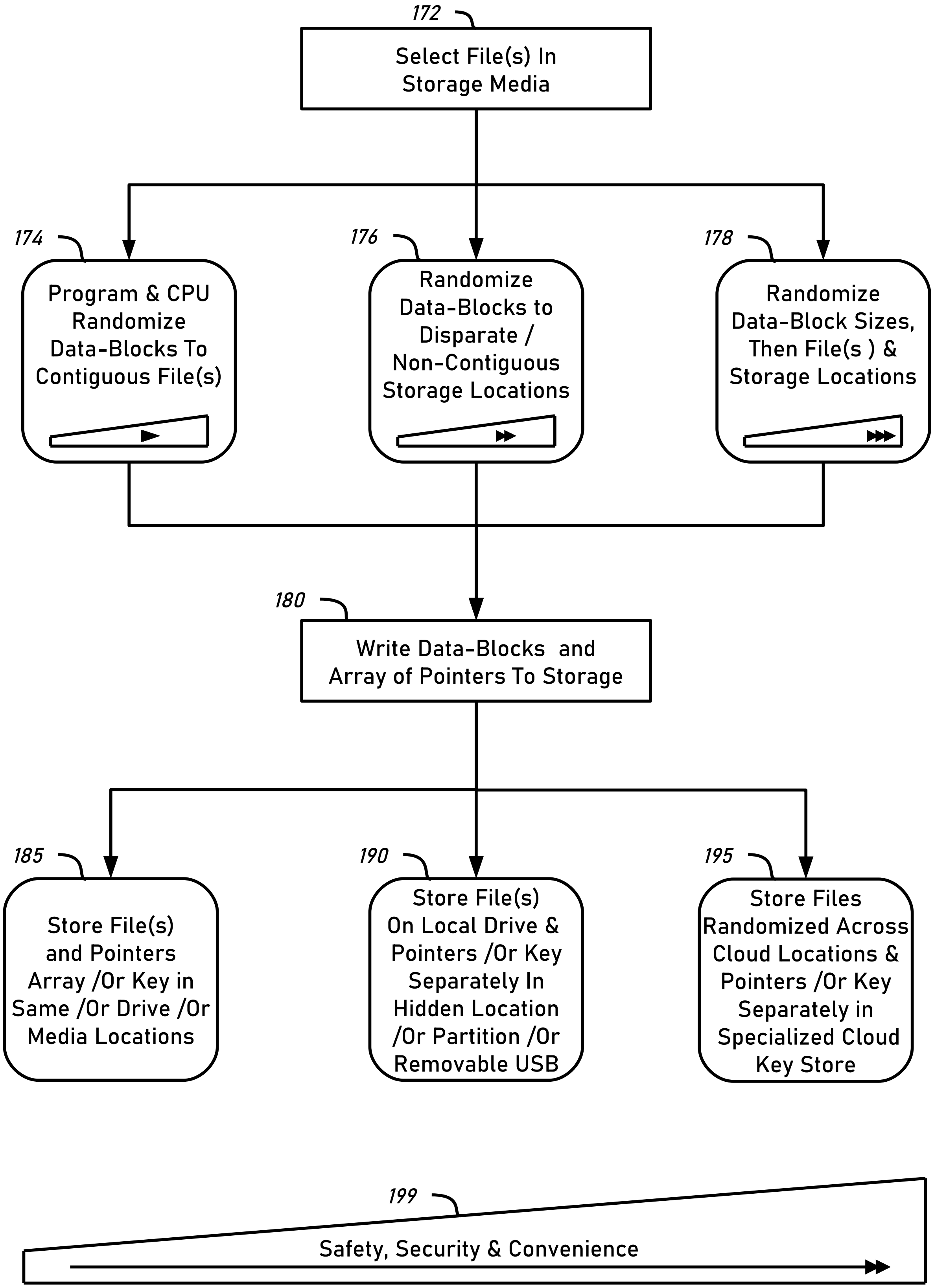
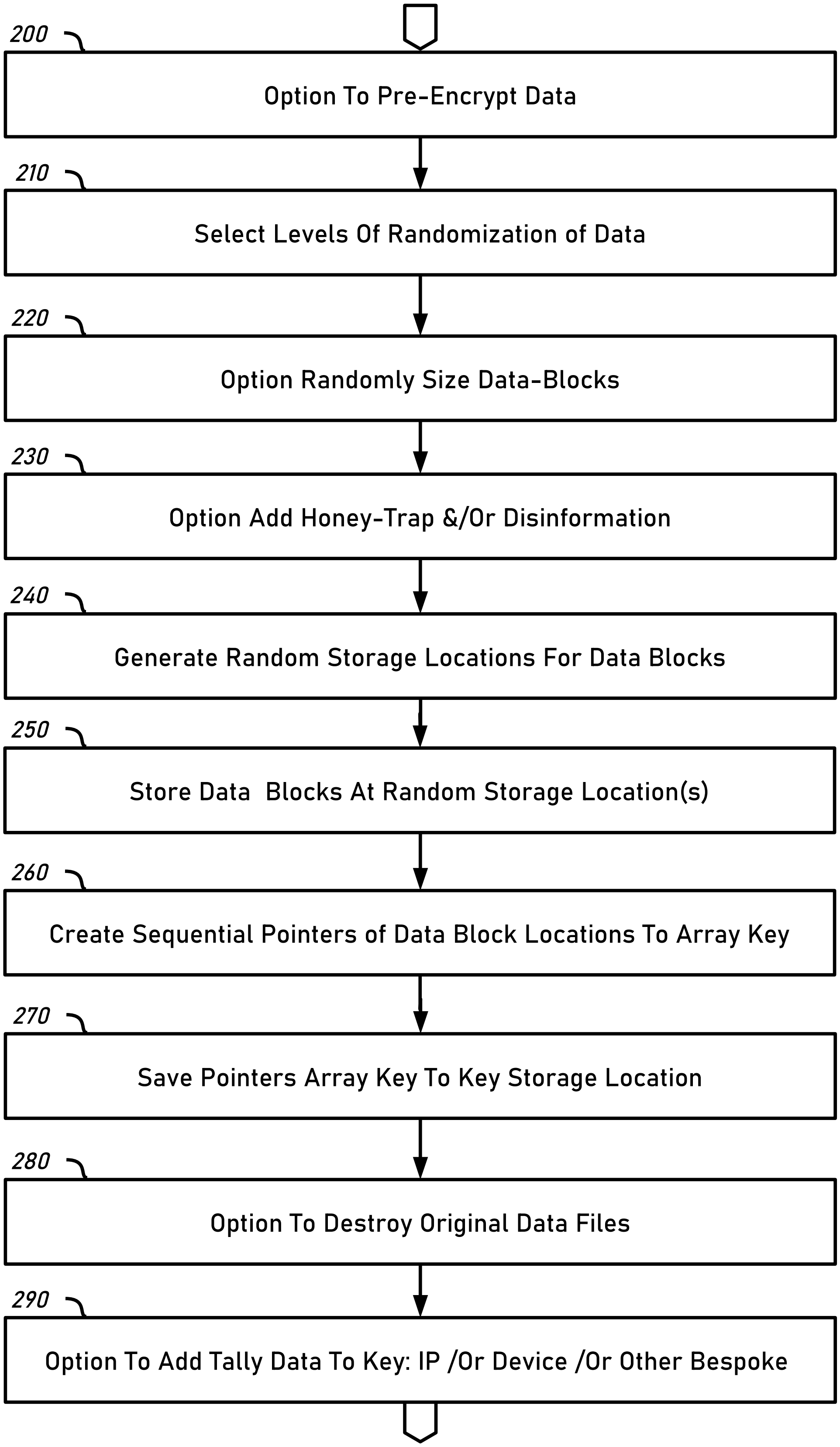

172
Select File(s) In Storage Media
174
Program & CPU Randomize Data-Blocks To Contiguous File(s)
176
Randomize Data-Blocks to Disparate / Non-Contiguous Storage Locations
178
Randomize Data-Block Sizes, Then File(s) & Storage Locations
180
Write Data-Blocks and Array of Pointers To Storage
185
Store File(s) and Pointers Array /Or Key in Same /Or Drive /Or Media Locations
190
Store File(s) On Local Drive & Pointers /Or Key Separately In Hidden Location /Or Partition /Or Removable USB
195
Store Files Randomized Across Cloud Locations & Pointers /Or Key Separately in Specialized Cloud Key Store
199
Safety, Security & Convenience

FIG. 24


200
Option To Pre-Encrypt Data
210
Select Levels Of Randomization of Data
220
Option Randomly Size Data-Blocks
230
Option Add Honey-Trap &/Or Disinformation
240
Generate Random Storage Locations For Data Blocks
250
Store Data Blocks At Random Storage Location(s)
260
Create Sequential Pointers of Data Block Locations To Array Key
270
Save Pointers Array Key To Key Storage Location
280
Option To Destroy Original Data Files
290
Option To Add Tally Data To Key: IP /Or Device /Or Other Bespoke

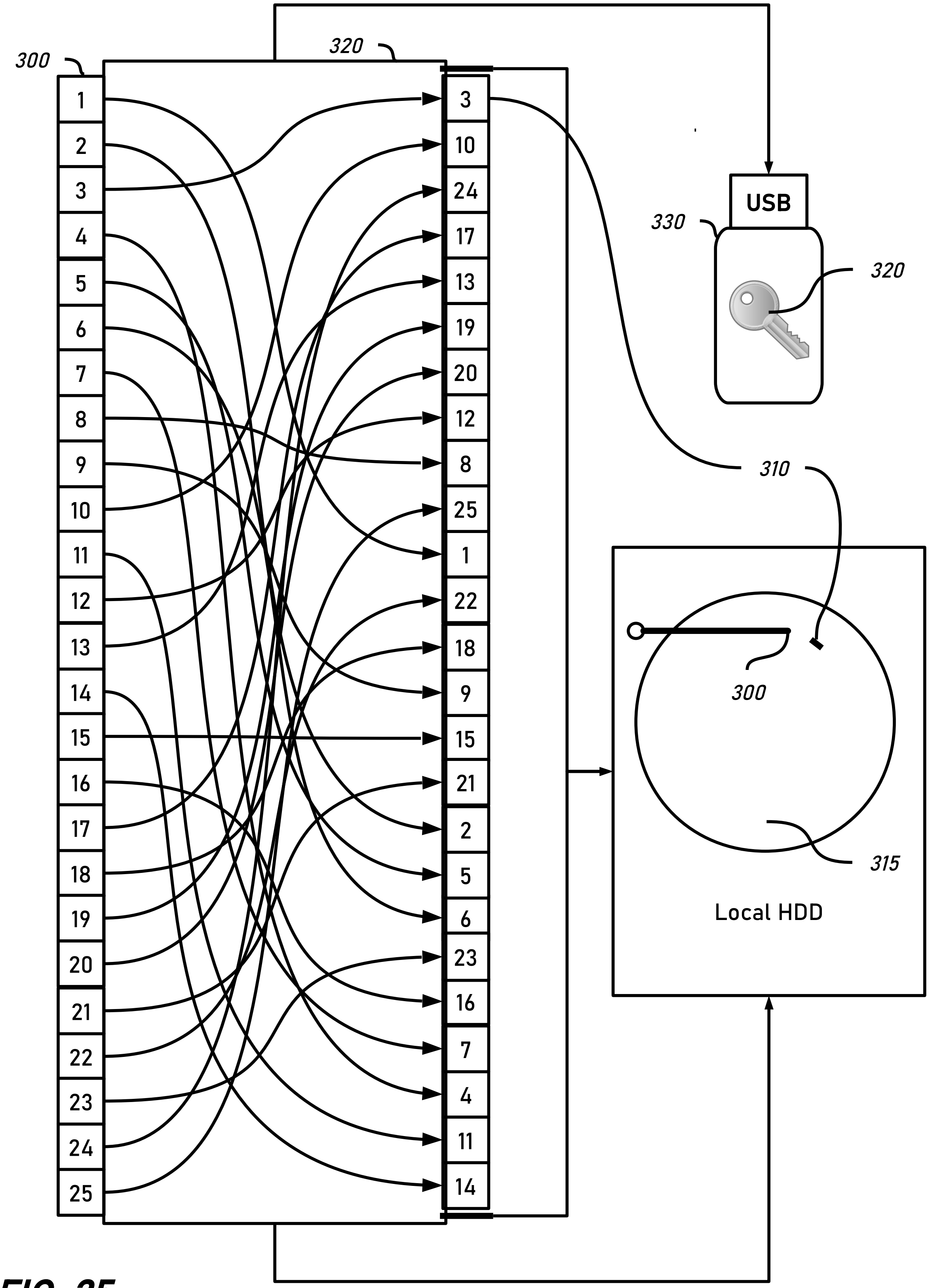

300
320
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
3
10
24
17
13
19
20
12
8
25
1
22
18
9
15
21
2
5
6
23
16
7
4
11
14
USB
330
320
310
300
315
Local HDD

FIG. 25

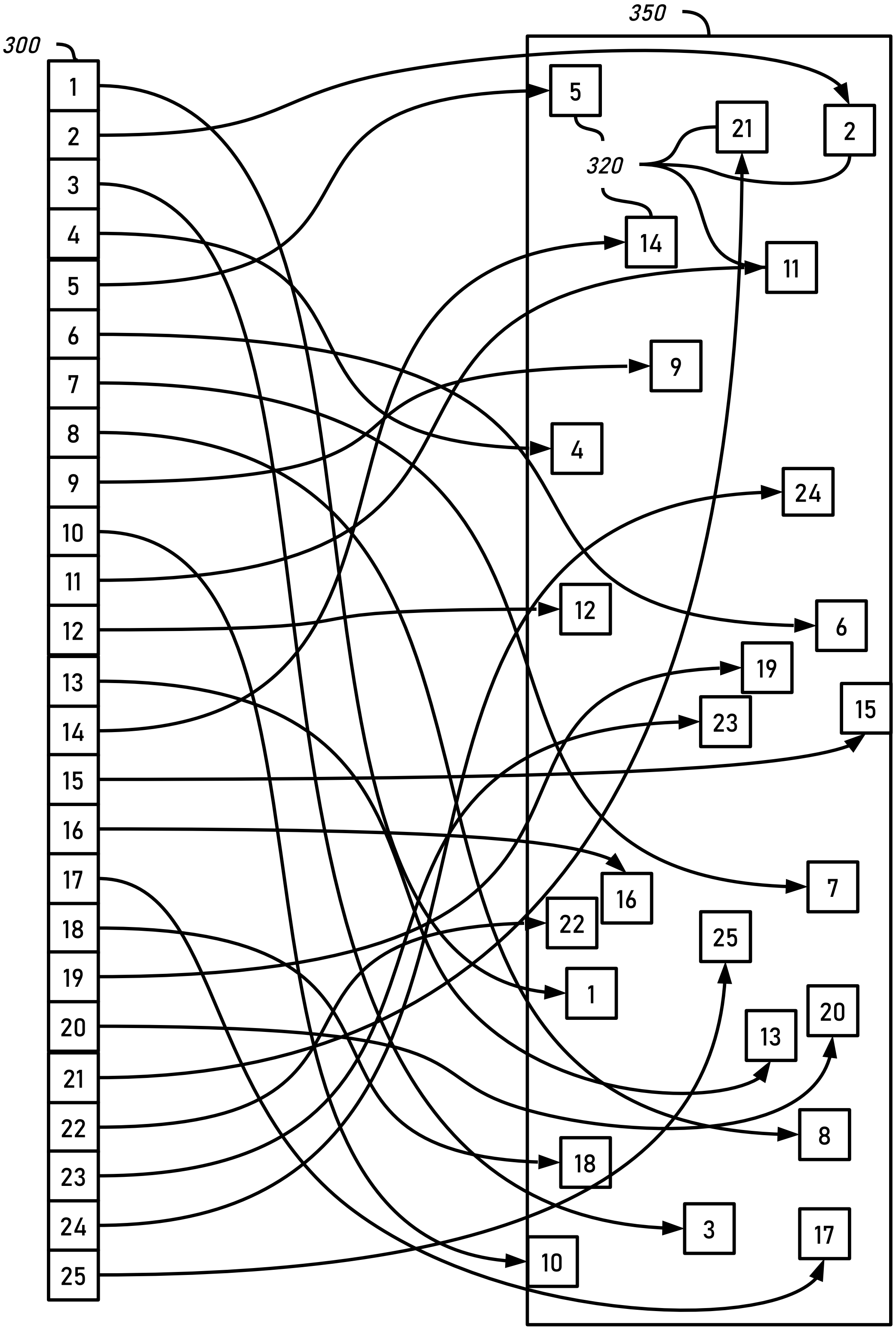

300
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
350
5
21
2
320
14
11
9
4
24
12
6
19
23
15
7
16
22
25
1
20
13
8
18
3
17
10

FIG. 26

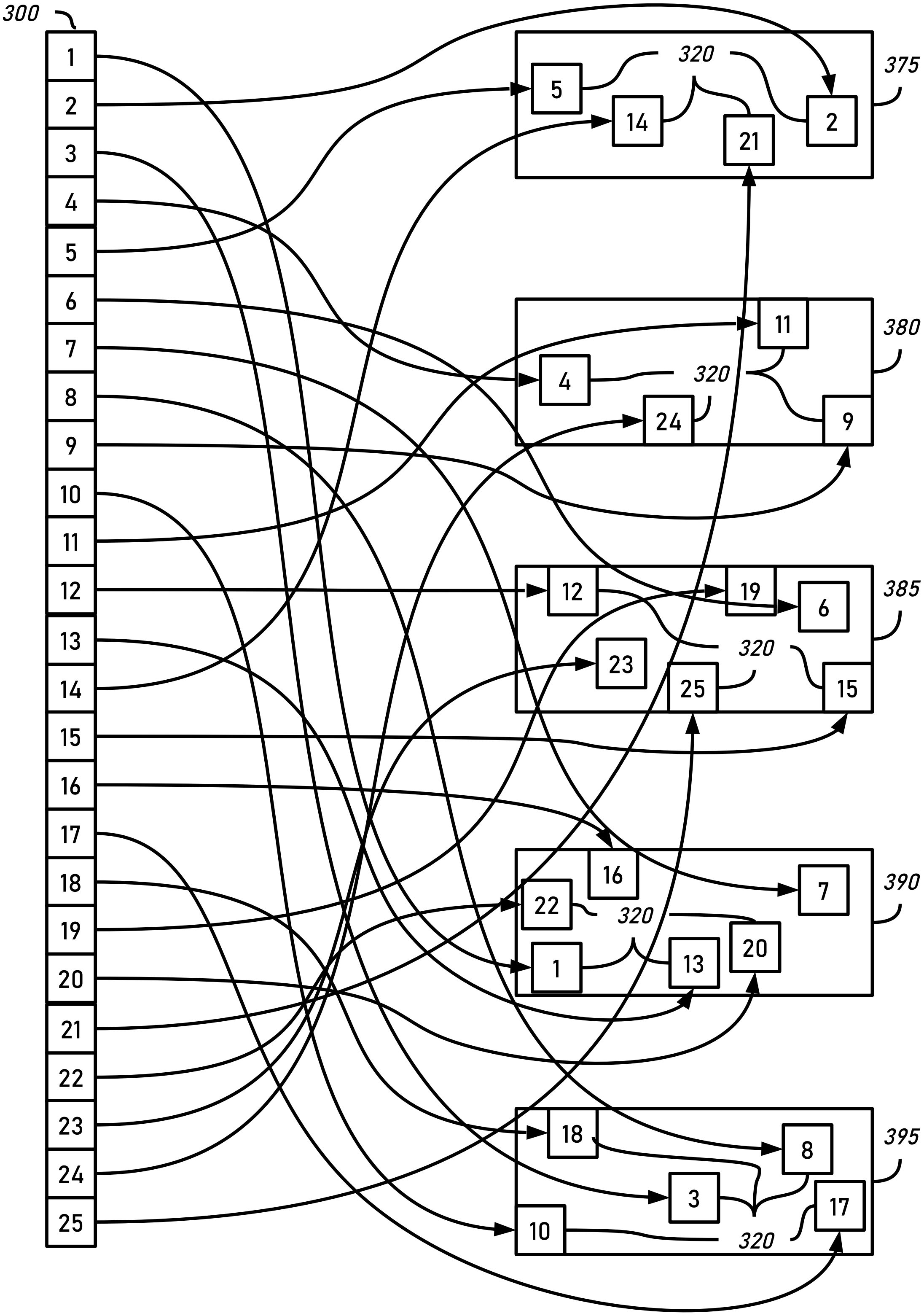

300
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
375
5
14
320
21
2
380
11
4
320
24
9
385
12
19
6
23
320
25
15
390
16
7
22
320
1
13
20
395
18
8
3
17
10
320

FIG. 27

FIG. 28

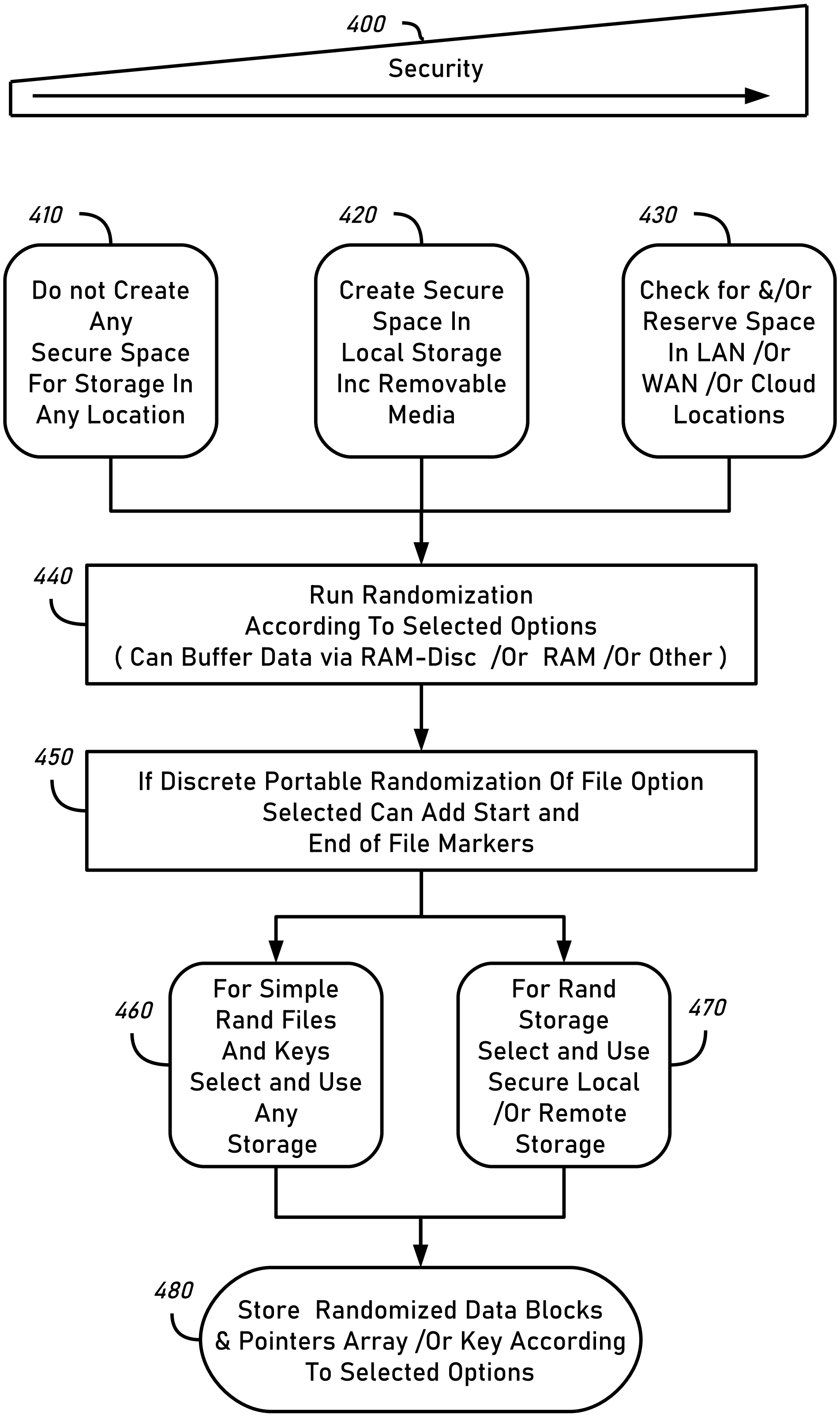

400
Security
410
Do not Create Any Secure Space For Storage In Any Location
420
Create Secure Space In Local Storage Inc Removable Media
430
Check for &/Or Reserve Space In LAN /Or WAN /Or Cloud Locations
440
Run Randomization According To Selected Options (Can Buffer Data via RAM-Disc /Or RAM /Or Other)
450
If Discrete Portable Randomization Of File Option Selected Can Add Start and End of File Markers
460
For Simple Rand Files And Keys Select and Use Any Storage
470
For Rand Storage Select and Use Secure Local /Or Remote Storage
480
Store Randomized Data Blocks & Pointers Array /Or Key According To Selected Options

FIG. 29

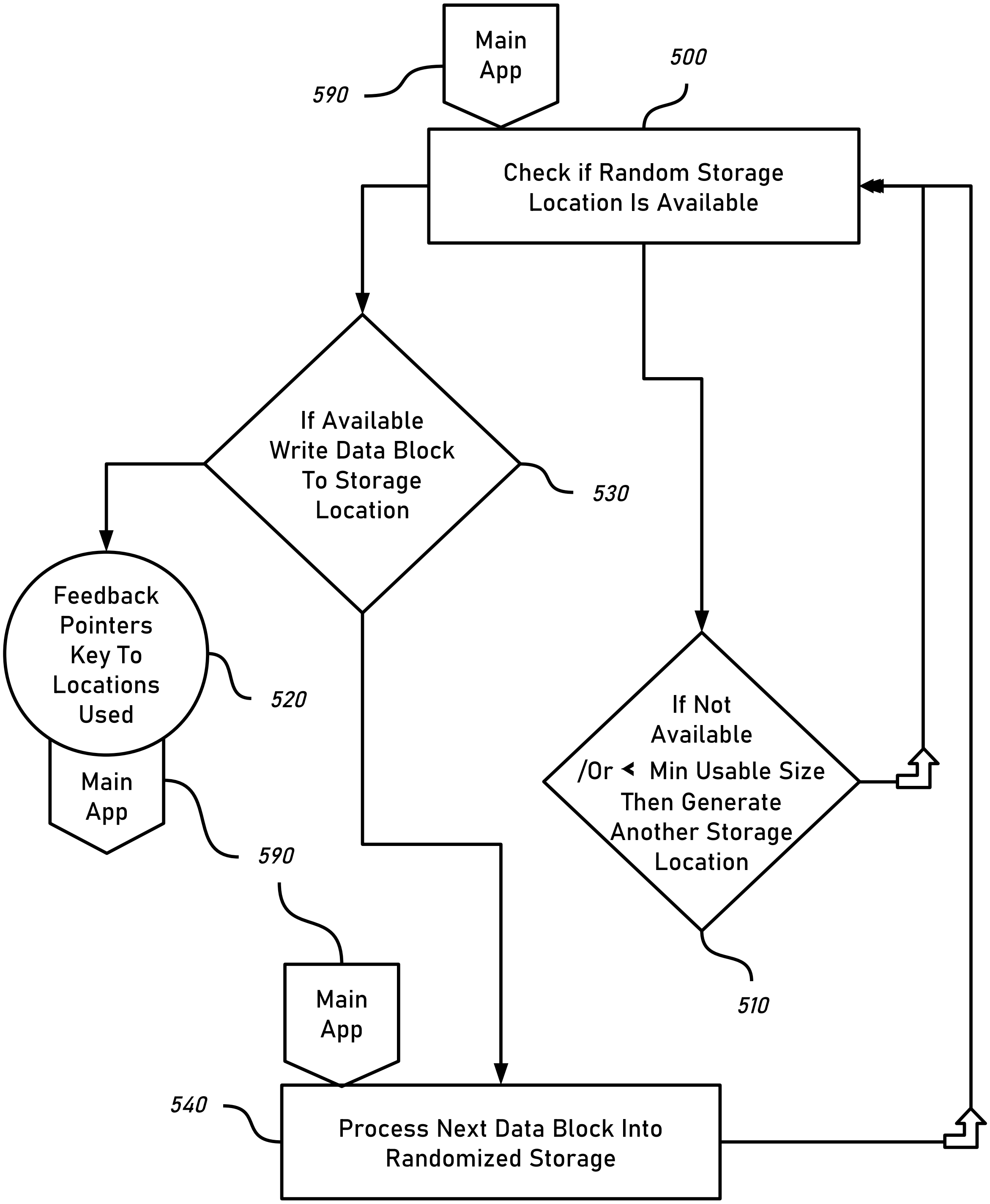

Main App
590
500
Check if Random Storage Location Is Available
If Available Write Data Block To Storage Location
530
Feedback Pointers Key To Locations Used
520
Main App
590
If Not Available /Or < Min Usable Size Then Generate Another Storage Location
590
510
Main App
540
Process Next Data Block Into Randomized Storage

FIG. 30

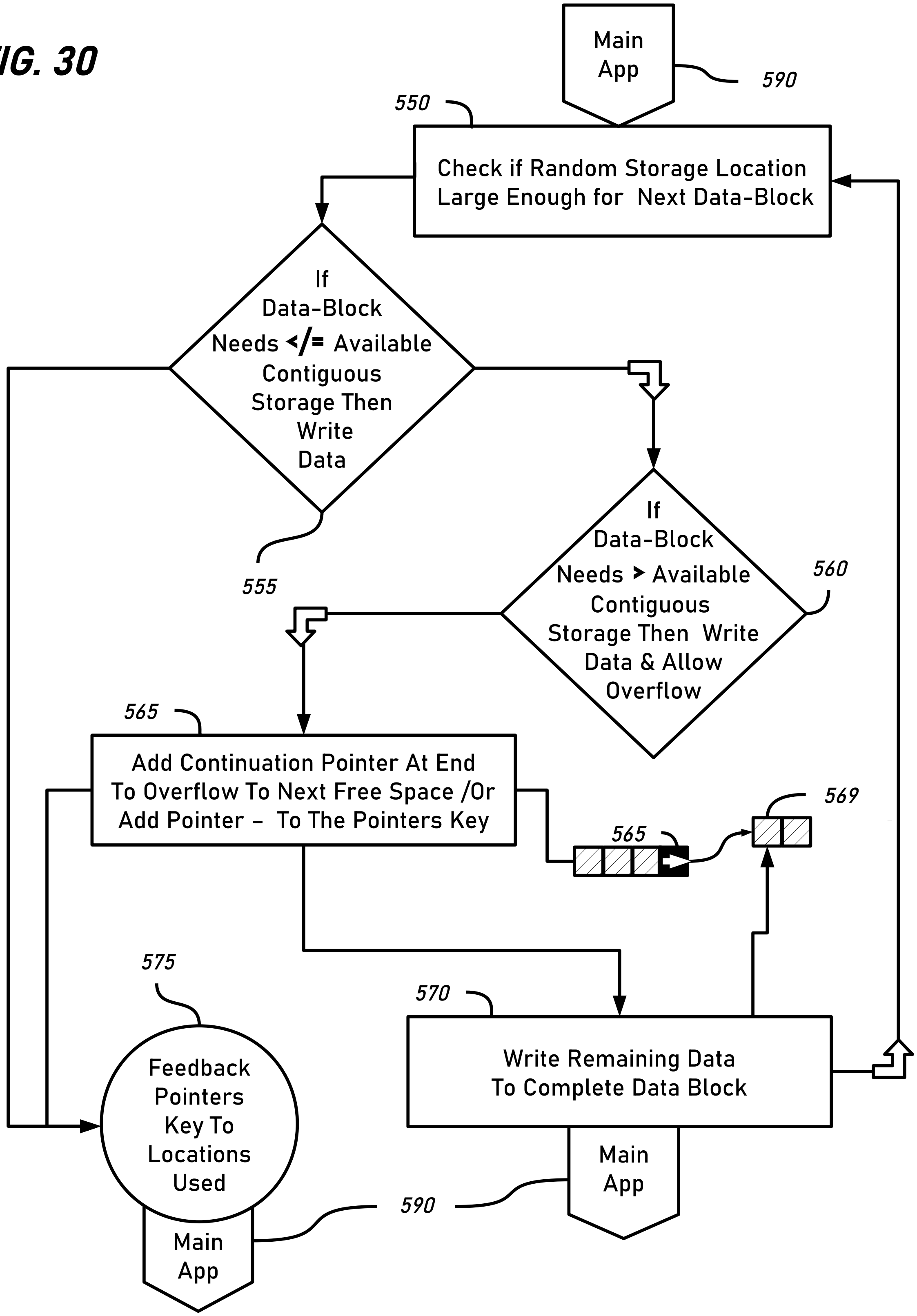

Main App
590
550
Check if Random Storage Location Large Enough for Next Data-Block
If Data-Block Needs </= Available Contiguous Storage Then Write Data
555
If Data-Block Needs > Available Contiguous Storage Then Write Data & Allow Overflow
560
565
Add Continuation Pointer At End To Overflow To Next Free Space /Or Add Pointer – To The Pointers Key
565
569
575
Feedback Pointers Key To Locations Used
570
Write Remaining Data To Complete Data Block
Main App
590
Main App

FIG. 31

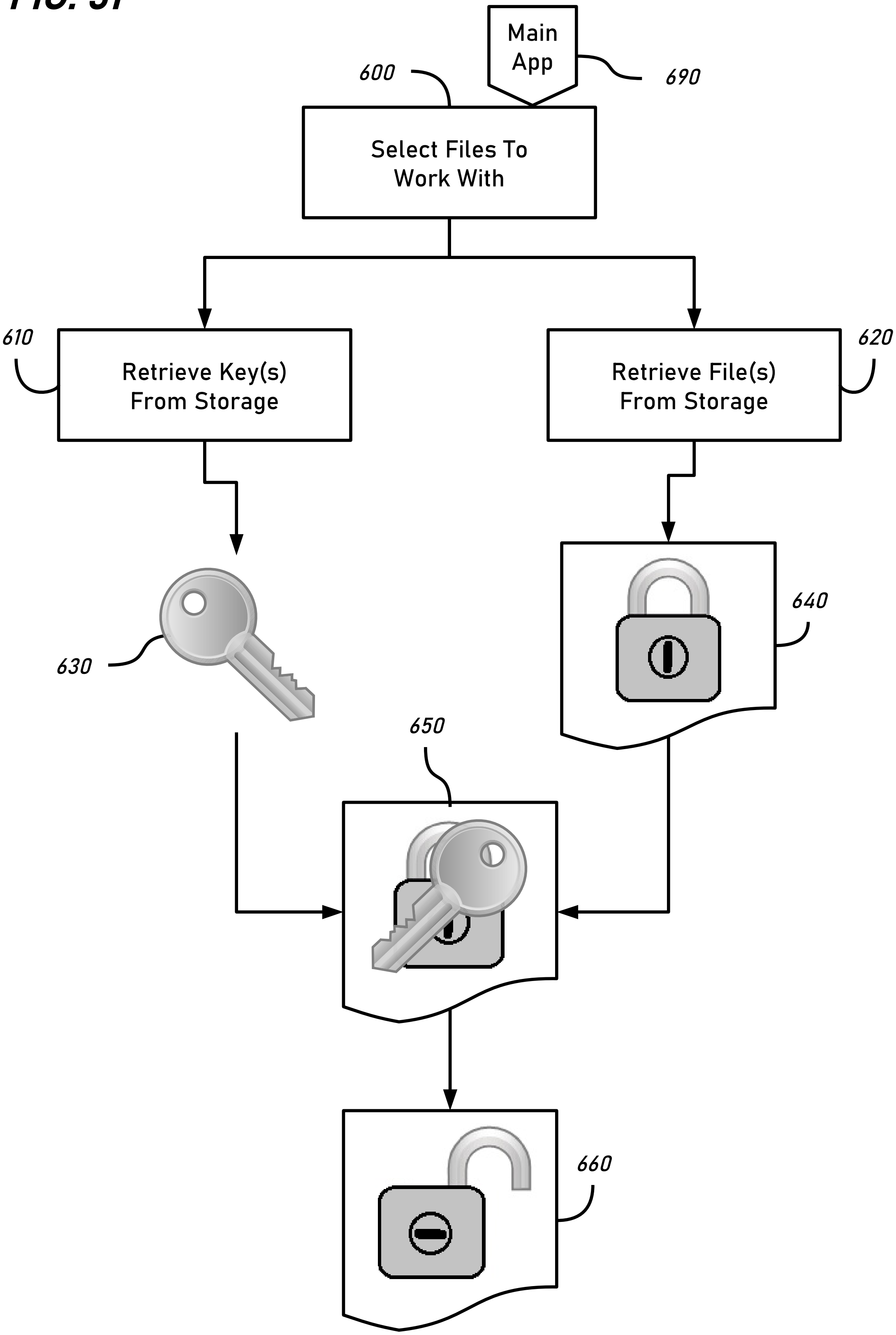

Main App
690
600
Select Files To Work With
610
Retrieve Key(s) From Storage
620
Retrieve File(s) From Storage
630
640
650
660

FIG. 32

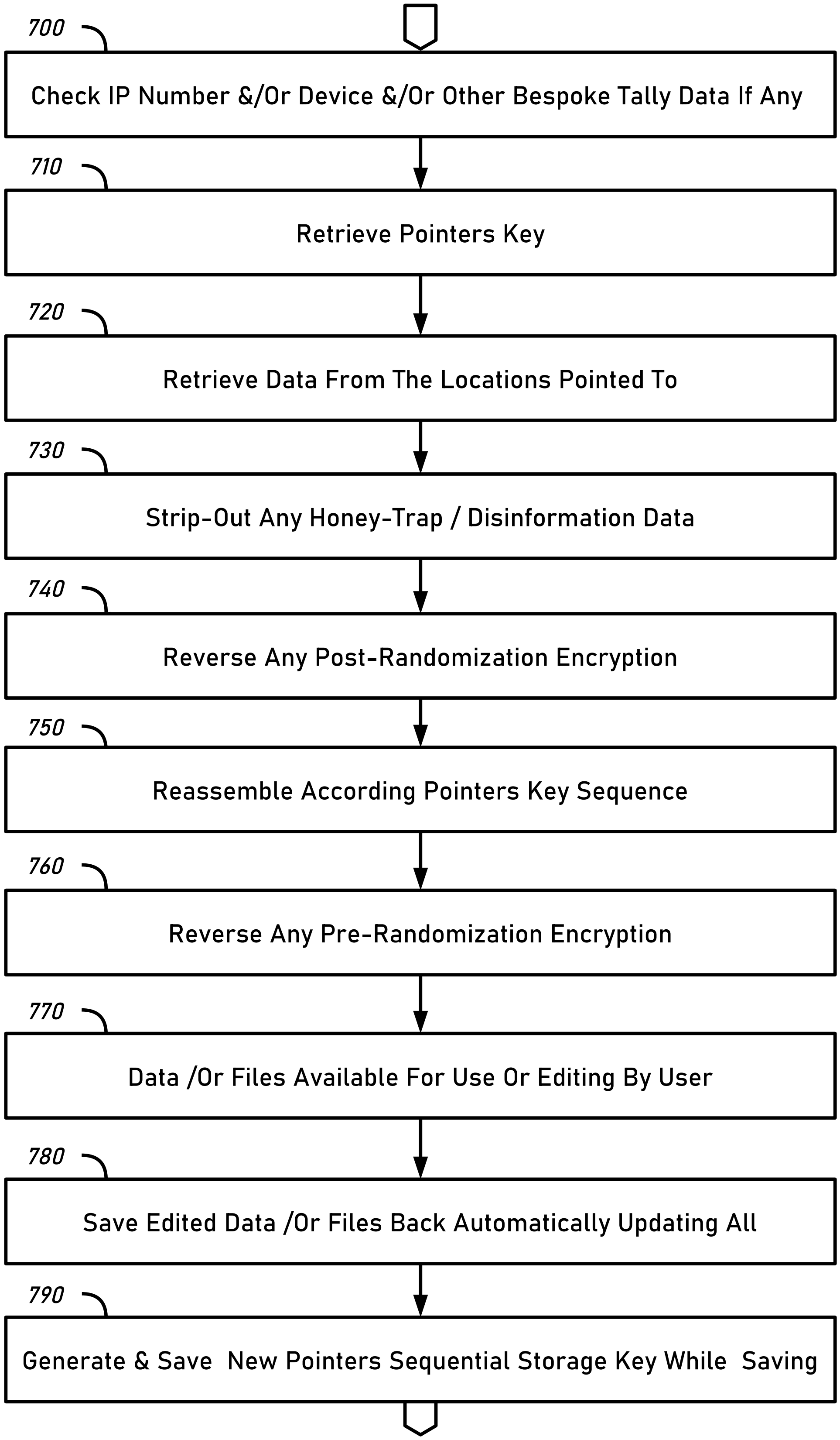

700
Check IP Number &/Or Device &/Or Other Bespoke Tally Data If Any
710
Retrieve Pointers Key
720
Retrieve Data From The Locations Pointed To
730
Strip-Out Any Honey-Trap / Disinformation Data
740
Reverse Any Post-Randomization Encryption
750
Reassemble According Pointers Key Sequence
760
Reverse Any Pre-Randomization Encryption
770
Data /Or Files Available For Use Or Editing By User
780
Save Edited Data /Or Files Back Automatically Updating All
790
Generate & Save New Pointers Sequential Storage Key While Saving

FIG. 33

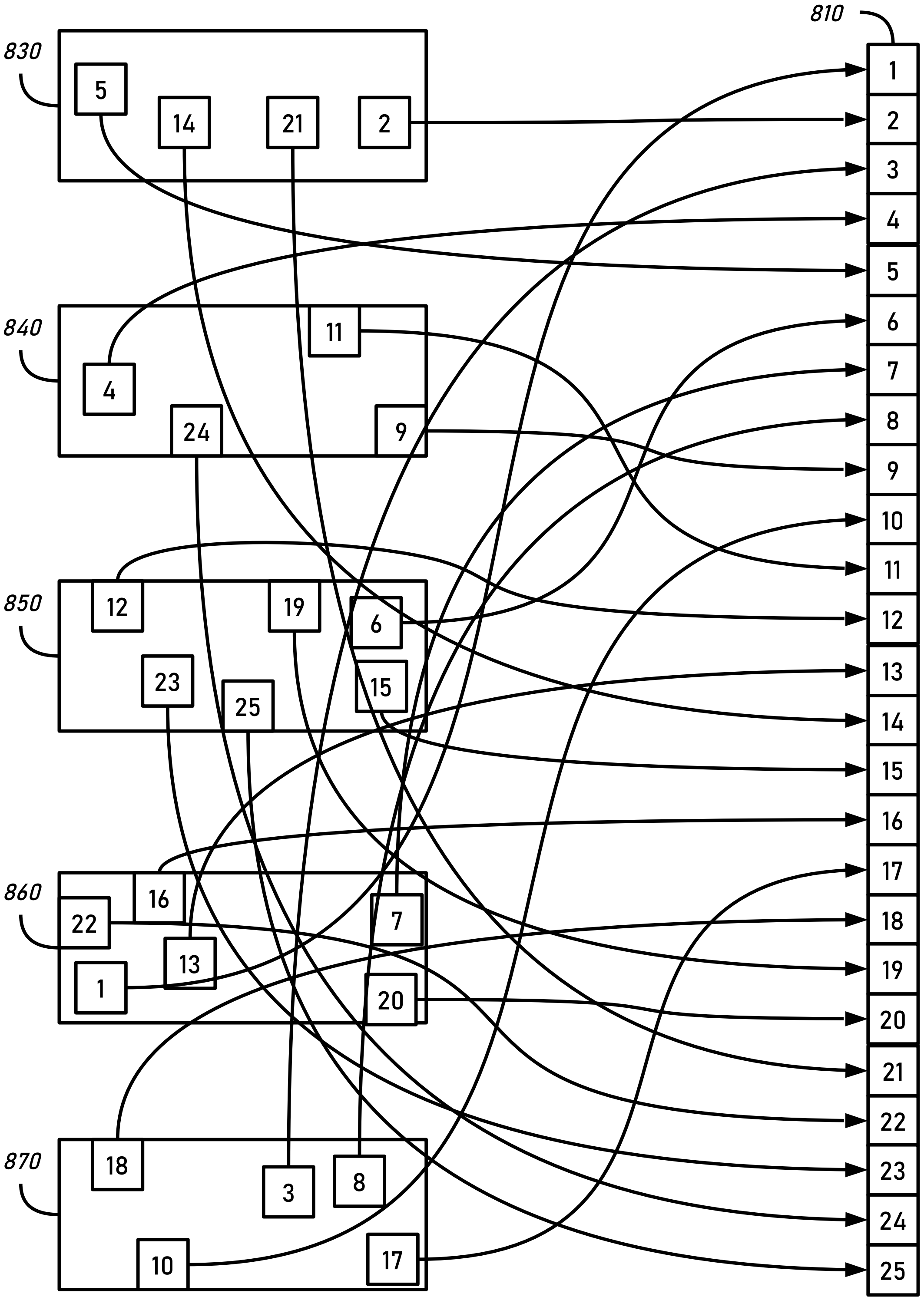

810
830
5
14
21
2
840
11
4
24
9
850
12
19
6
23
25
15
860
16
22
7
13
1
20
870
18
3
8
10
17
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

FIG. 34

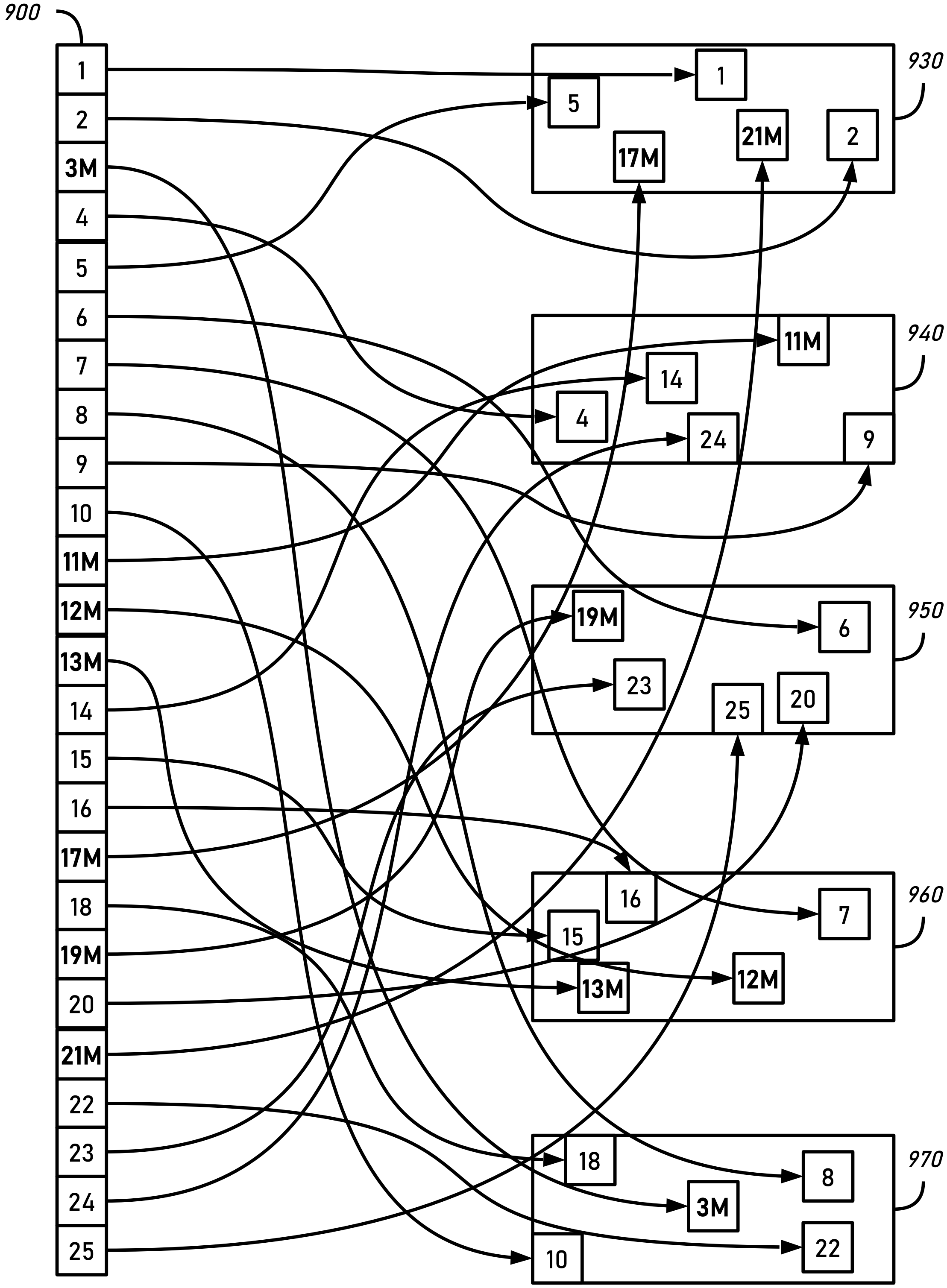

900
1
2
3M
4
5
6
7
8
9
10
11M
12M
13M
14
15
16
17M
18
19M
20
21M
22
23
24
25
930
1
5
17M
21M
2
940
11M
14
4
24
9
950
19M
6
23
25
20
960
16
7
15
12M
13M
970
18
8
3M
22
10

FIG. 35

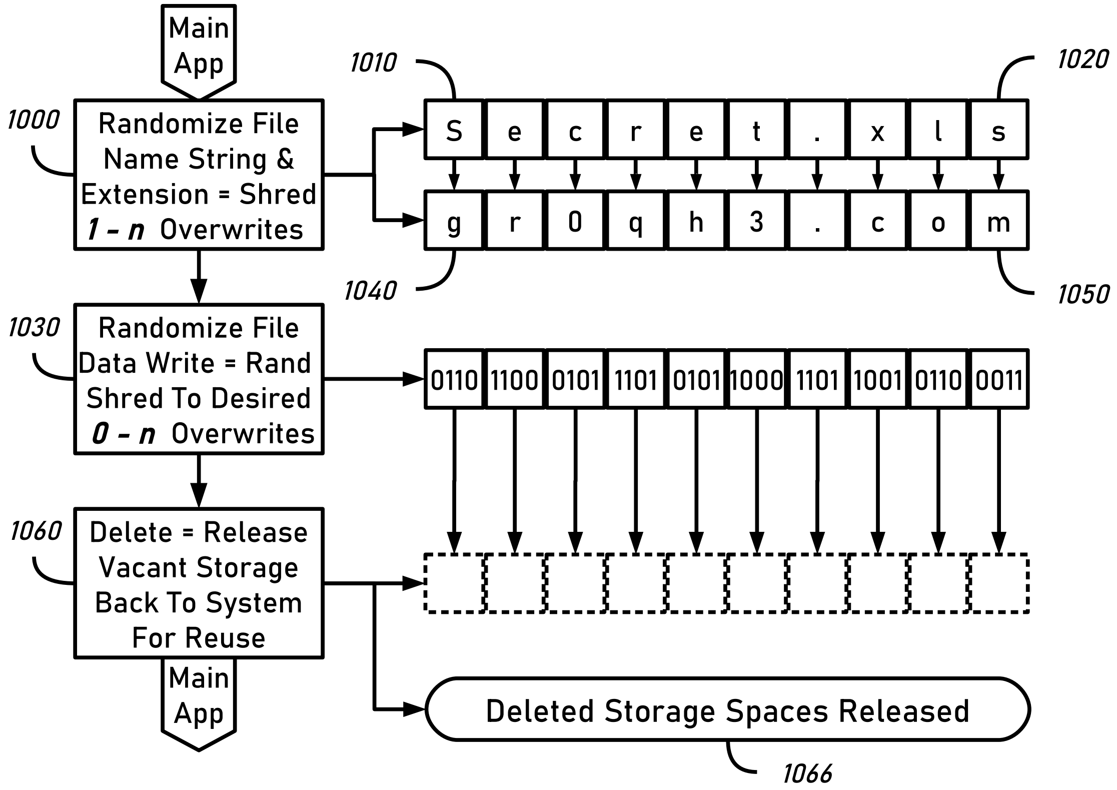

Main App
1000
Randomize File Name String & Extension = Shred 1 - n Overwrites
1010
1020
S e c r e t . x l s
g r 0 q h 3 . c o m
1040
1050
1030
Randomize File Data Write = Rand Shred To Desired 0 - n Overwrites
0110 1100 0101 1101 0101 1000 1101 1001 0110 0011
1060
Delete = Release Vacant Storage Back To System For Reuse
Main App
Deleted Storage Spaces Released
1066

FIG. 36

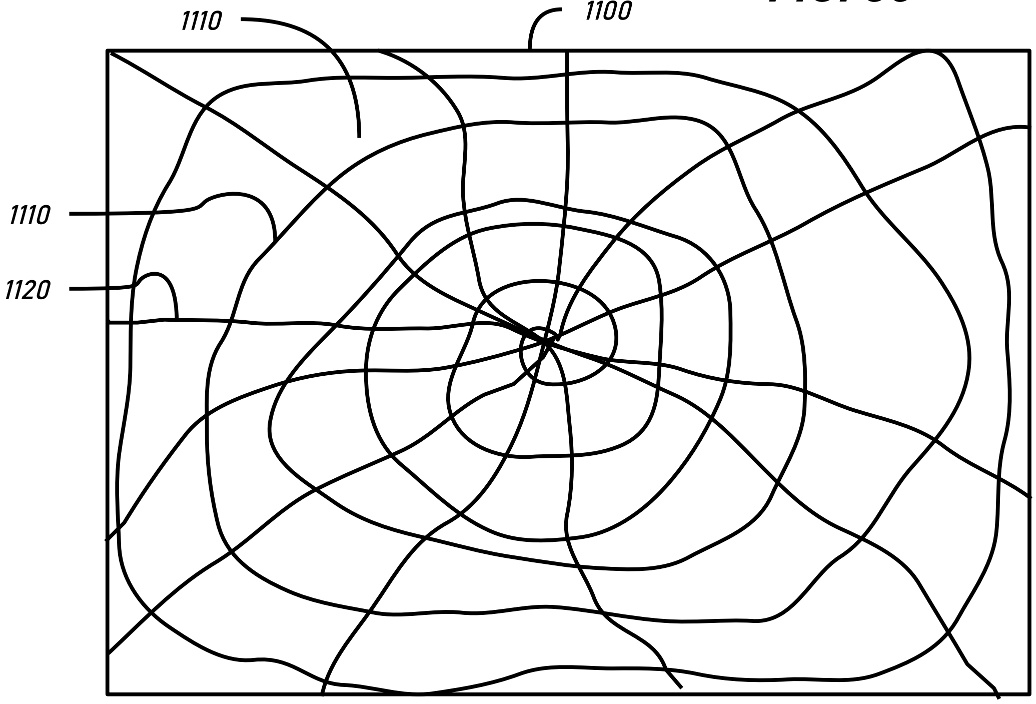

1110
1100
1110
1120

FIG. 37

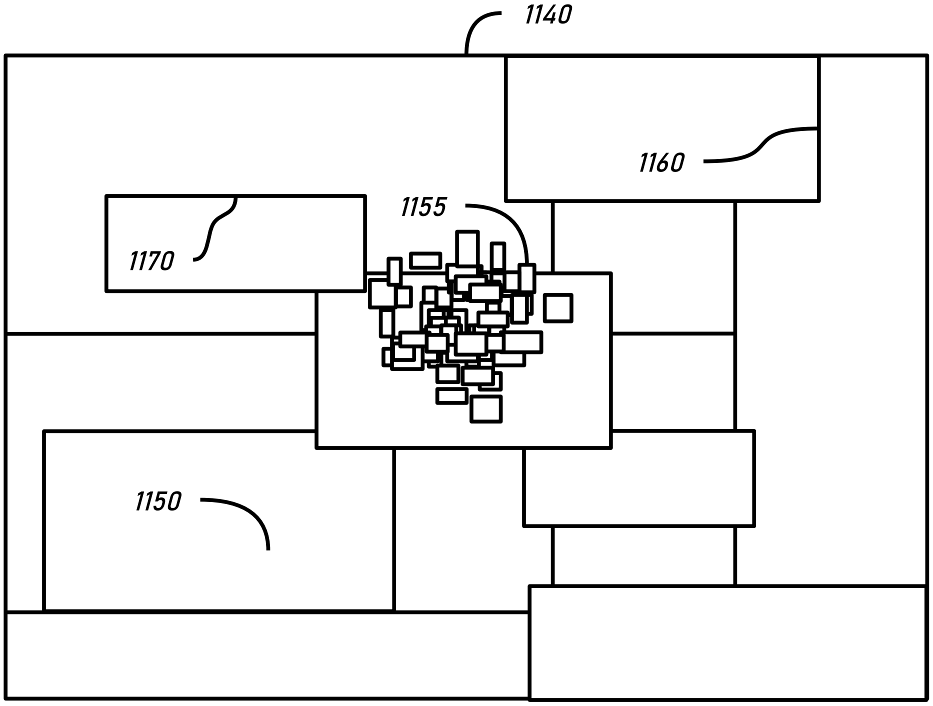


FIG. 38

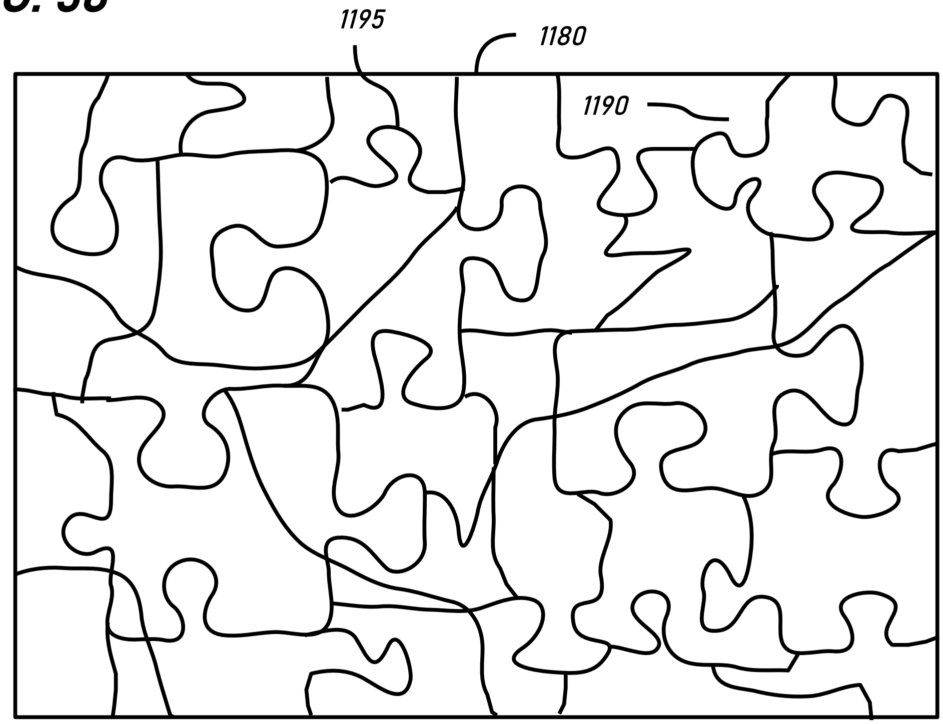
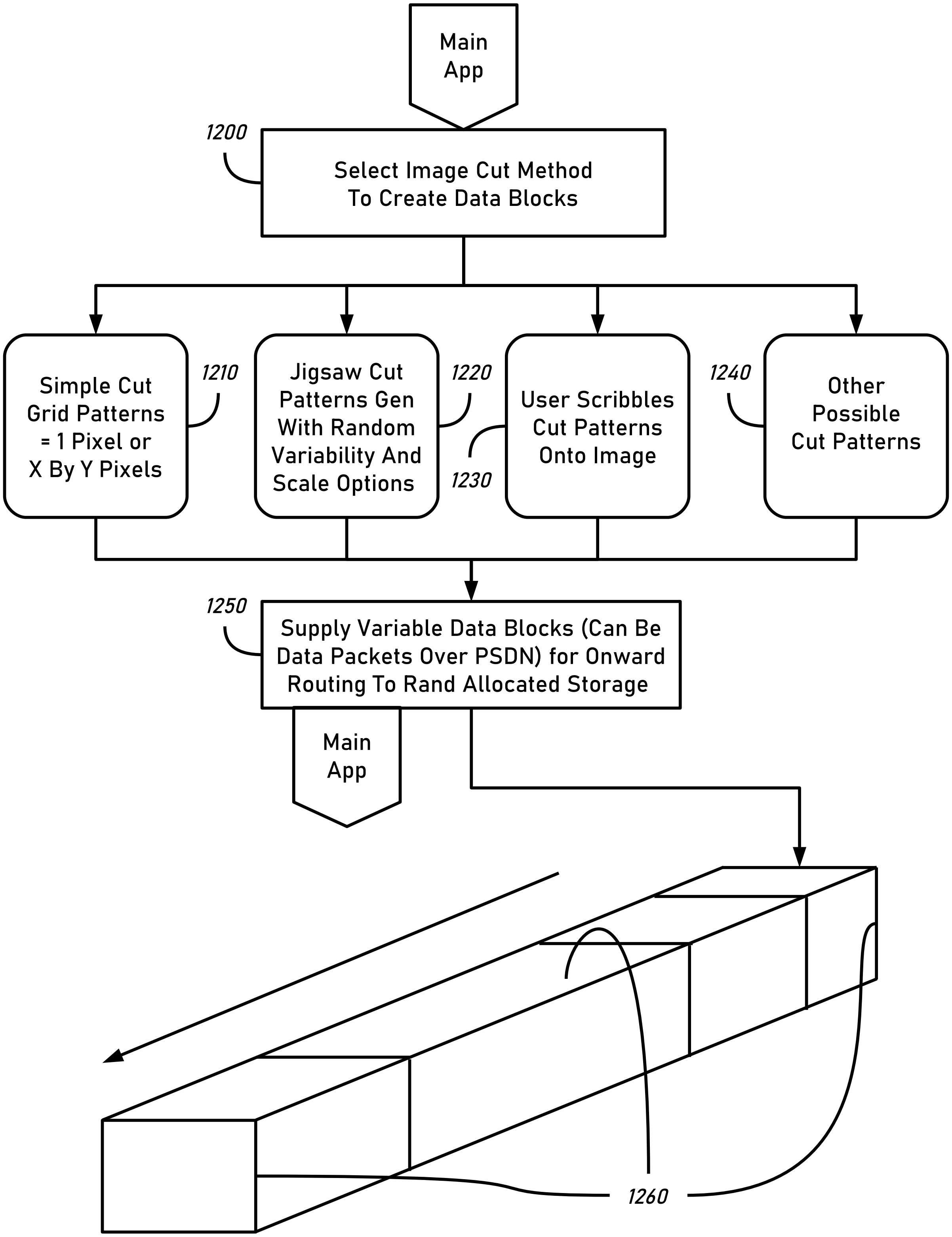


FIG. 39



Main App
1200
Select Image Cut Method To Create Data Blocks
Simple Cut Grid Patterns = 1 Pixel or X By Y Pixels
1210
Jigsaw Cut Patterns Gen With Random Variability And Scale Options
1220
User Scribbles Cut Patterns Onto Image
1230
1240
Other Possible Cut Patterns
1250
Supply Variable Data Blocks (Can Be Data Packets Over PSDN) for Onward Routing To Rand Allocated Storage
Main App
1260

FIG. 40

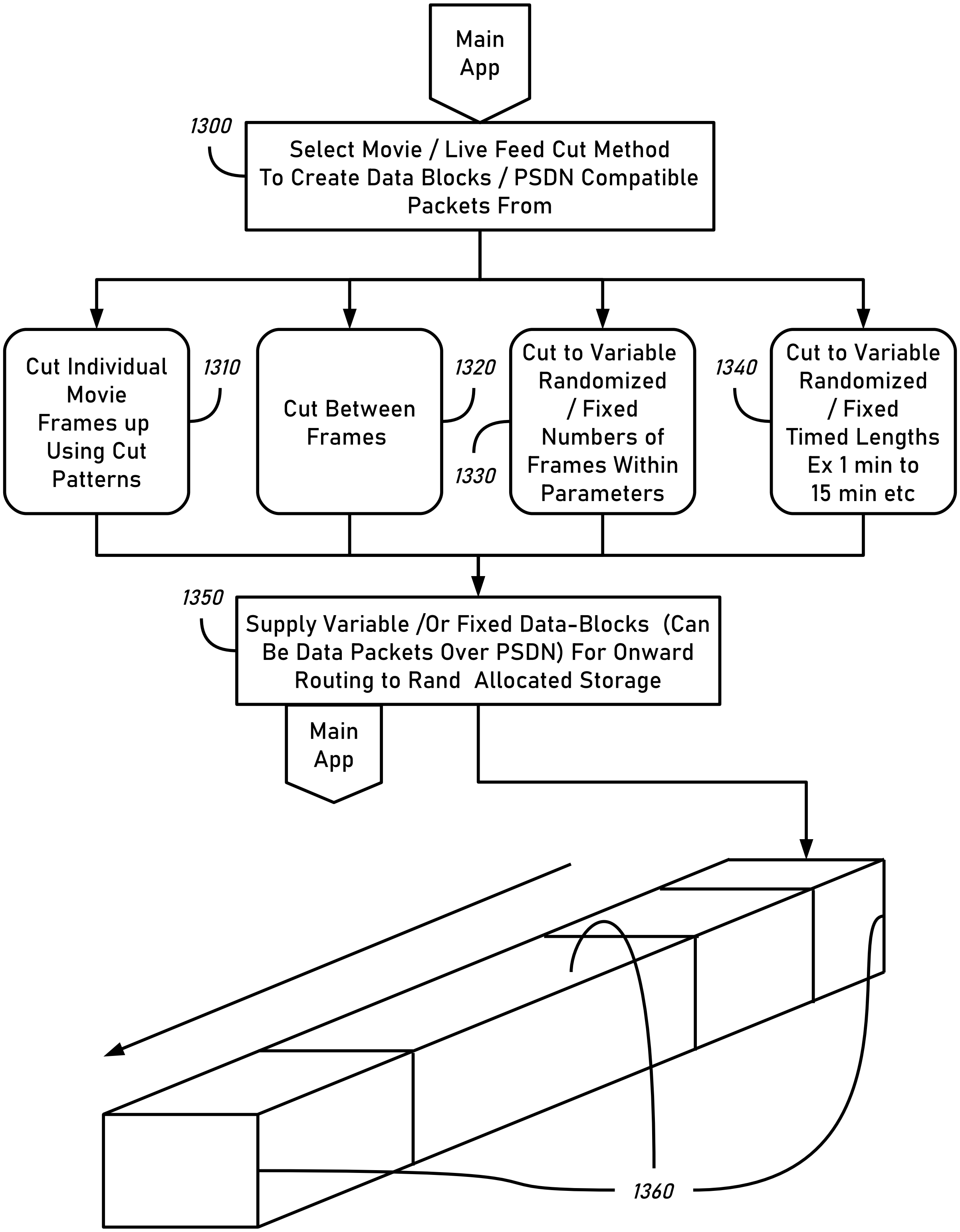

Main App
1300
Select Movie / Live Feed Cut Method To Create Data Blocks / PSDN Compatible Packets From
Cut Individual Movie Frames up Using Cut Patterns
1310
Cut Between Frames
1320
Cut to Variable Randomized / Fixed Numbers of Frames Within Parameters
1330
1340
Cut to Variable Randomized / Fixed Timed Lengths Ex 1 min to 15 min etc
1350
Supply Variable /Or Fixed Data-Blocks (Can Be Data Packets Over PSDN) For Onward Routing to Rand Allocated Storage
Main App
1360

FIG. 41

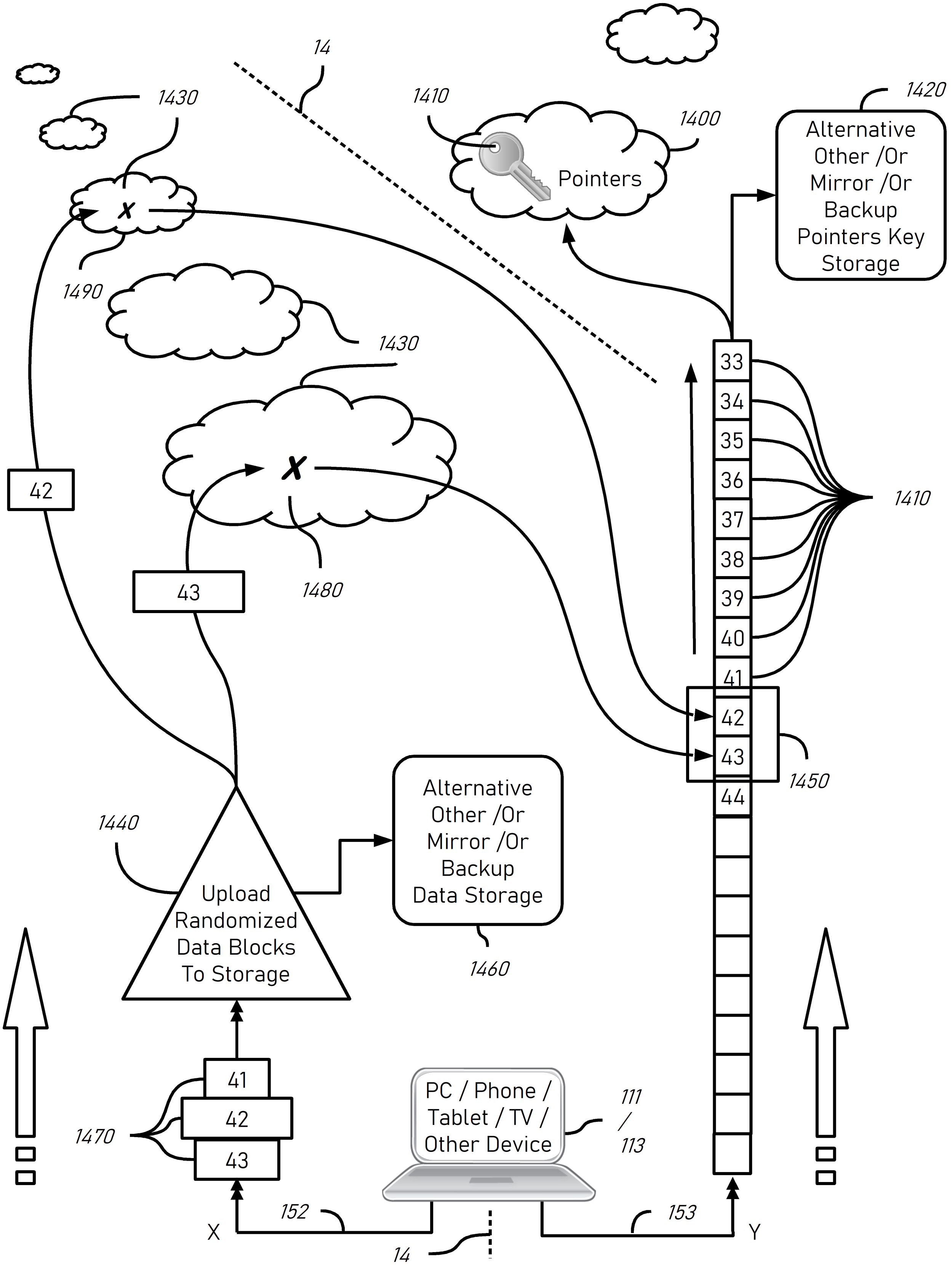

14
1430
1410
1400
Pointers
1420
Alternative Other /Or Mirror /Or Backup Pointers Key Storage
X
1490
1430
42
X
43
1480
33
34
35
36
37
38
39
40
41
42
43
44
1410
1450
1440
Upload Randomized Data Blocks To Storage
Alternative Other /Or Mirror /Or Backup Data Storage
1460
41
42
43
1470
PC / Phone / Tablet / TV / Other Device
111 / 113
152
153
X
Y
14

FIG. 41A

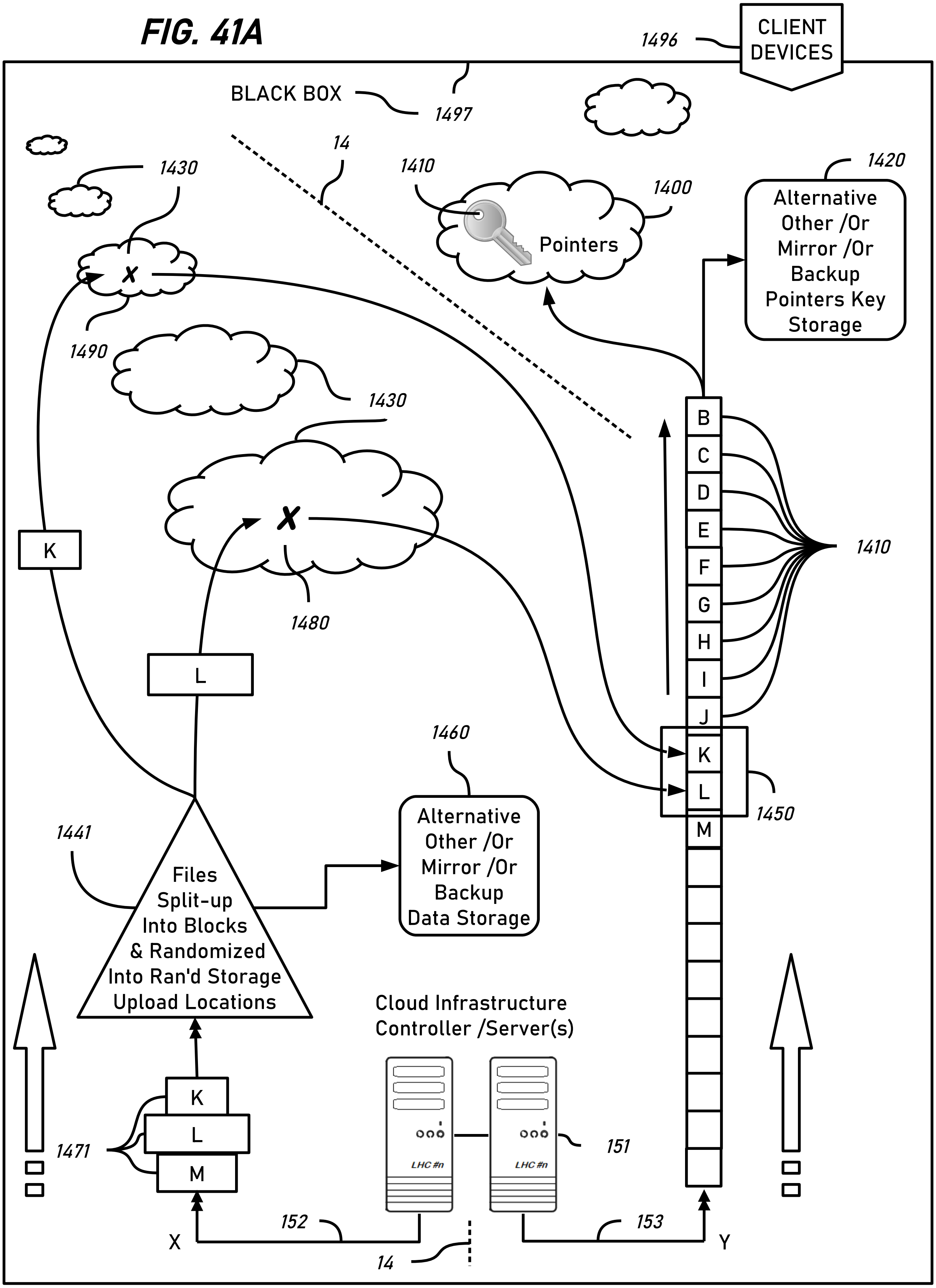

CLIENT DEVICES
1496
BLACK BOX
1497
14
1410
1400
Pointers
1420
Alternative Other /Or Mirror /Or Backup Pointers Key Storage
1430
1490
1430
1480
K
L
B
C
D
E
F
G
H
I
J
K
L
M
1410
1450
1460
Alternative Other /Or Mirror /Or Backup Data Storage
1441
Files Split-up Into Blocks & Randomized Into Ran'd Storage Upload Locations
Cloud Infrastructure Controller /Server(s)
LHC #n
LHC #n
151
1471
K
L
M
X
152
14
153
Y

FIG. 42

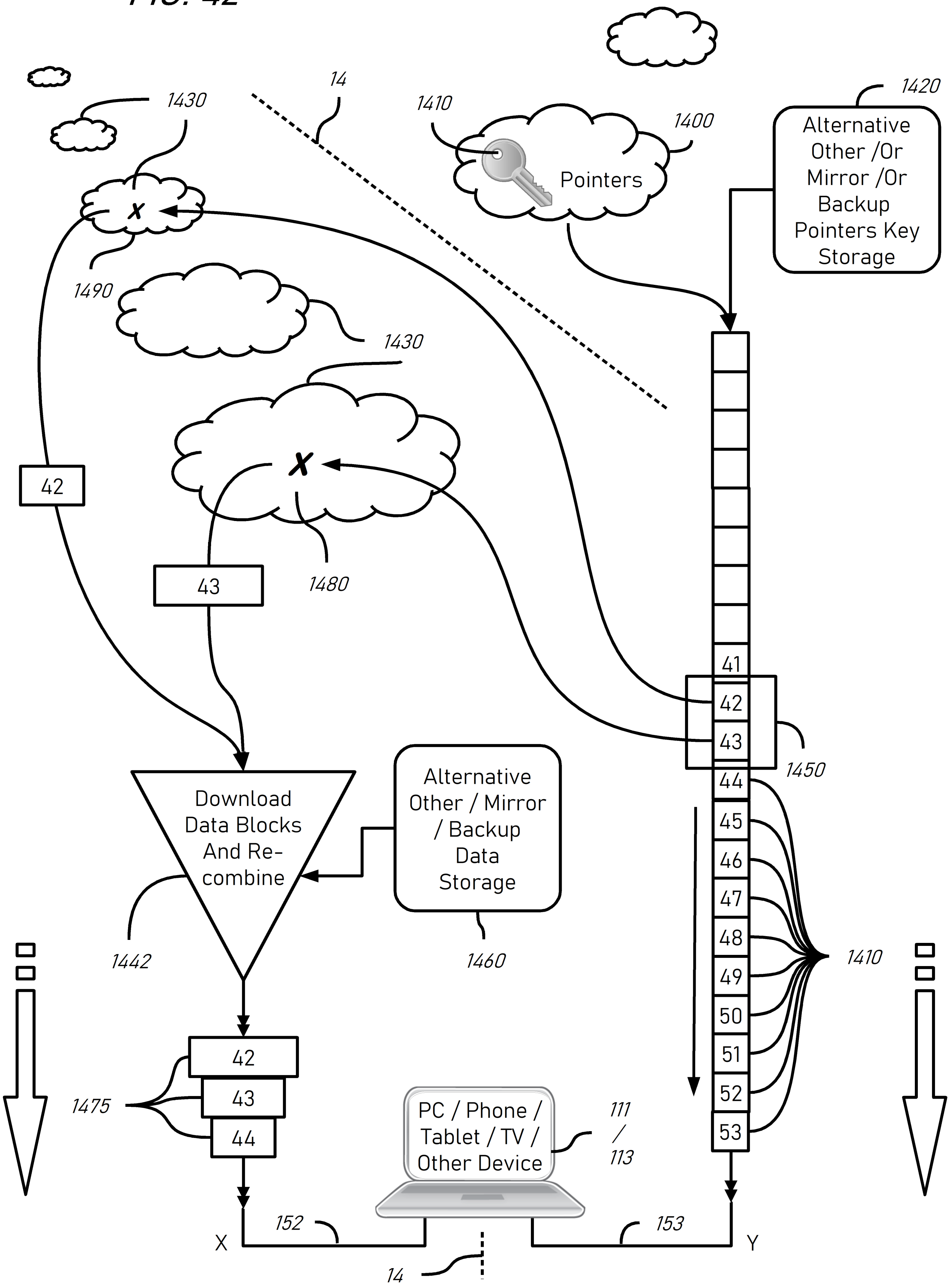

14
1430
1410
1400
1420
Alternative Other /Or Mirror /Or Backup Pointers Key Storage
Pointers
x
1490
1430
X
42
43
1480
41
42
43
1450
44
45
46
47
48
49
50
51
52
53
1410
Download Data Blocks And Re-combine
Alternative Other / Mirror / Backup Data Storage
1442
1460
42
43
44
1475
PC / Phone / Tablet / TV / Other Device
111 / 113
152
153
X
Y
14

FIG. 42A

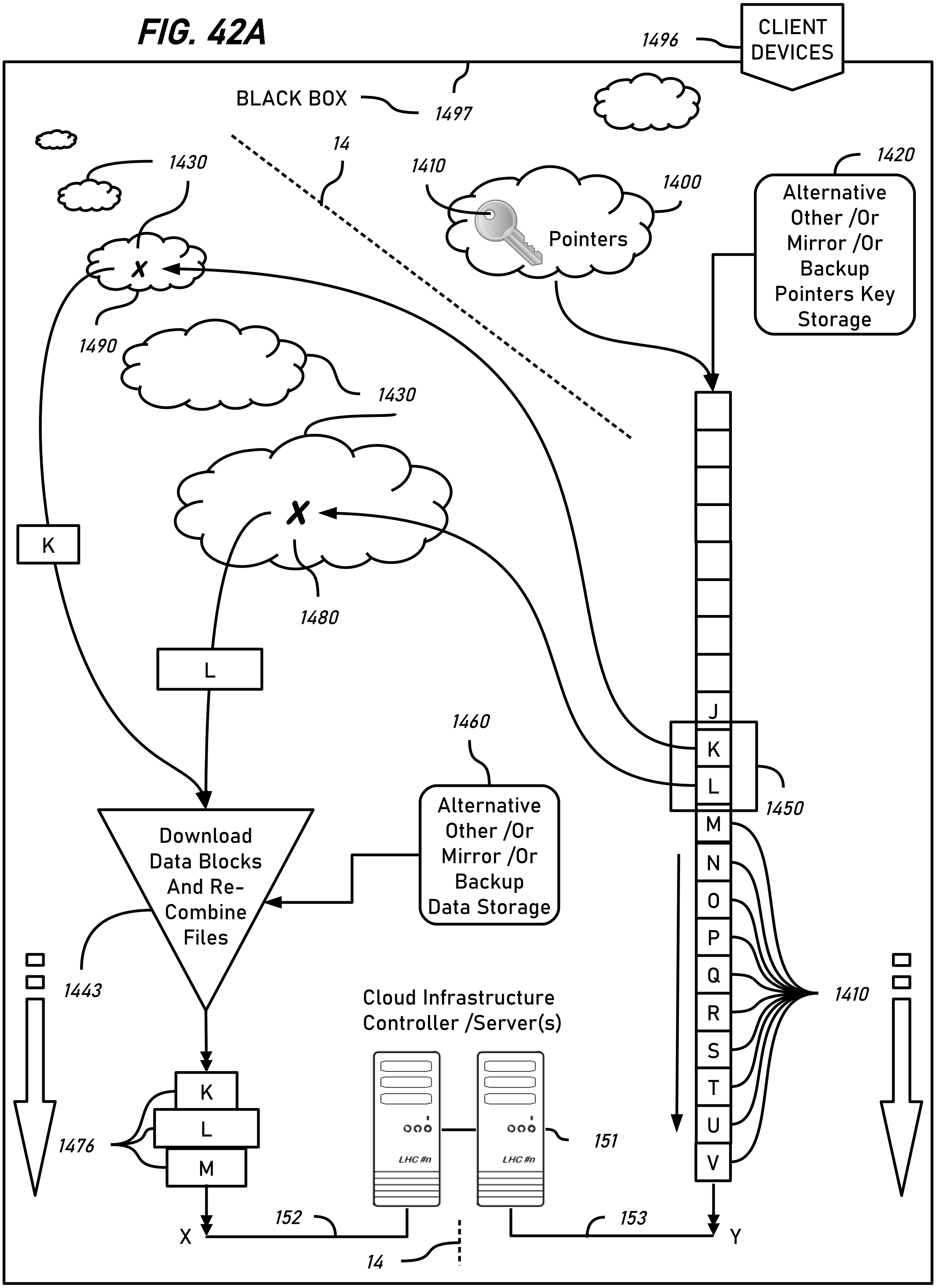

CLIENT DEVICES
1496
BLACK BOX
1497
14
1410
1400
Pointers
1420
Alternative Other /Or Mirror /Or Backup Pointers Key Storage
1430
1490
1430
K
L
1480
J
K
L
1450
M
N
O
P
Q
R
S
T
U
V
1410
1460
Alternative Other /Or Mirror /Or Backup Data Storage
Download Data Blocks And Re-Combine Files
1443
Cloud Infrastructure Controller /Server(s)
LHC #n
LHC #n
151
K
L
M
1476
152
153
X
Y
14

FIG. 43

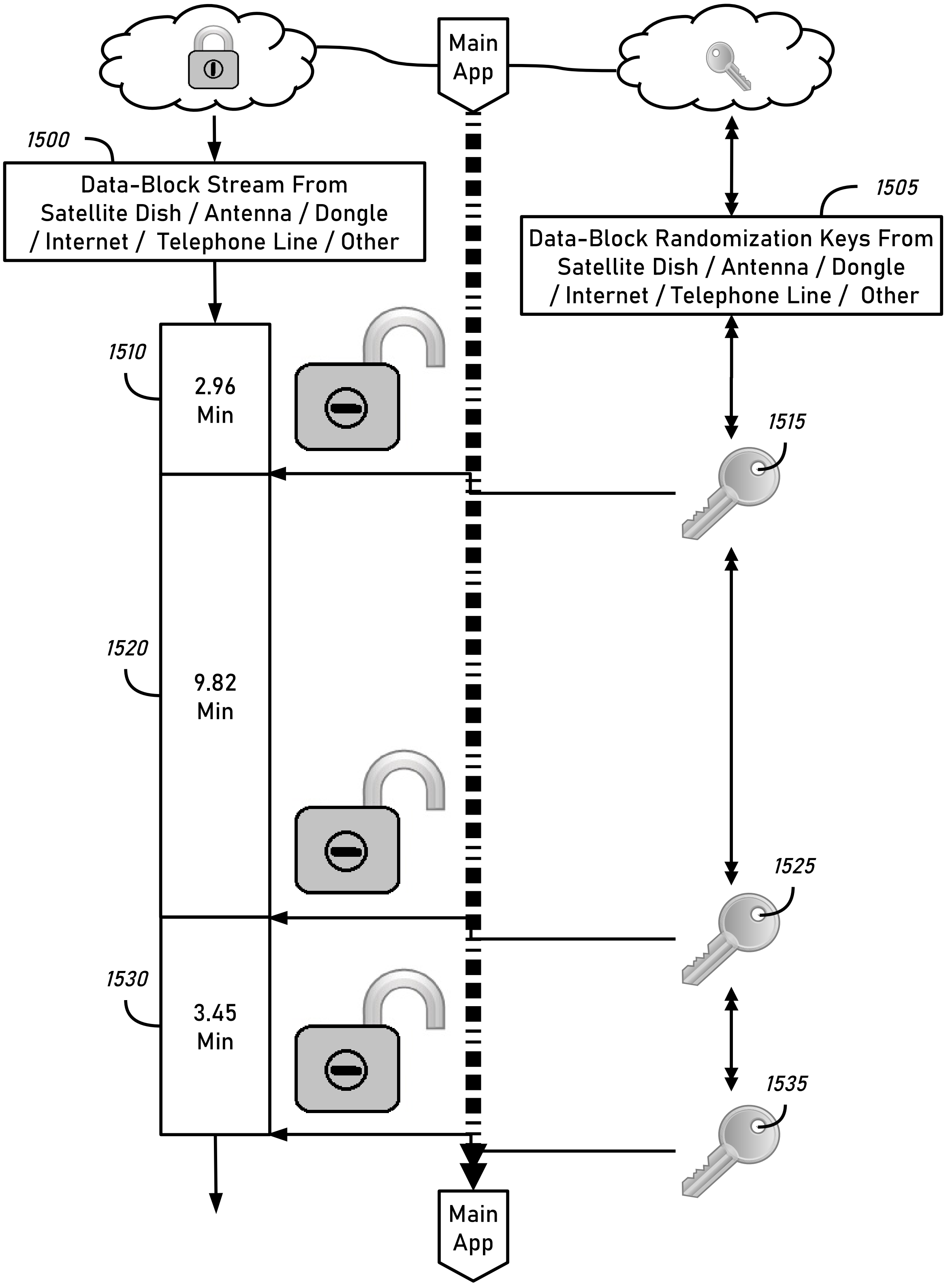

Main
App
1500
Data-Block Stream From
Satellite Dish / Antenna / Dongle
/ Internet / Telephone Line / Other
1505
Data-Block Randomization Keys From
Satellite Dish / Antenna / Dongle
/ Internet / Telephone Line / Other
1510
2.96
Min
1515
1520
9.82
Min
1525
1530
3.45
Min
1535
Main
App

FIG. 44

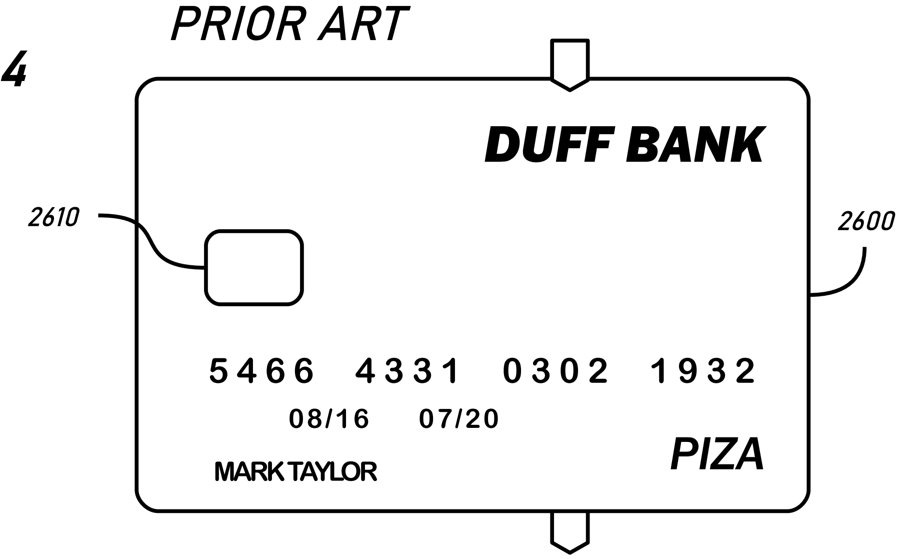


FIG. 45

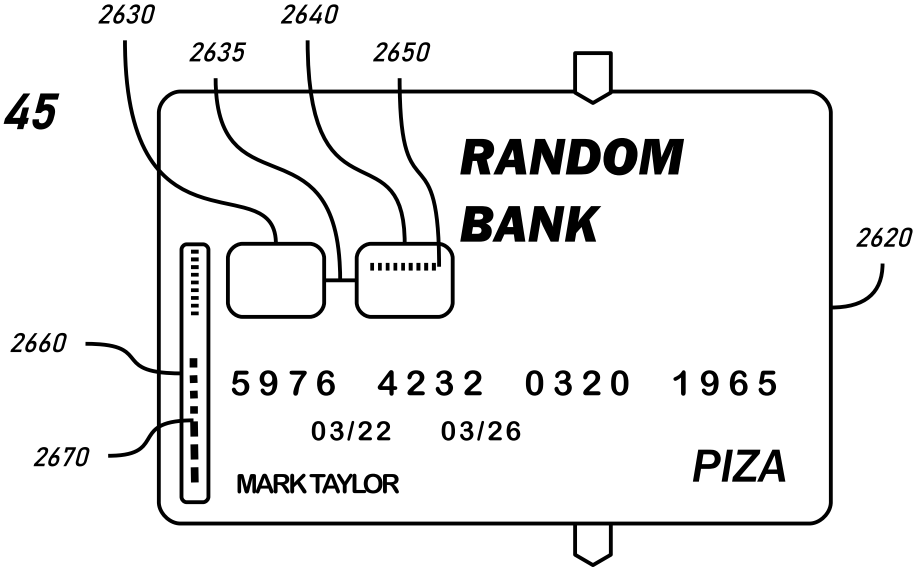


FIG. 46

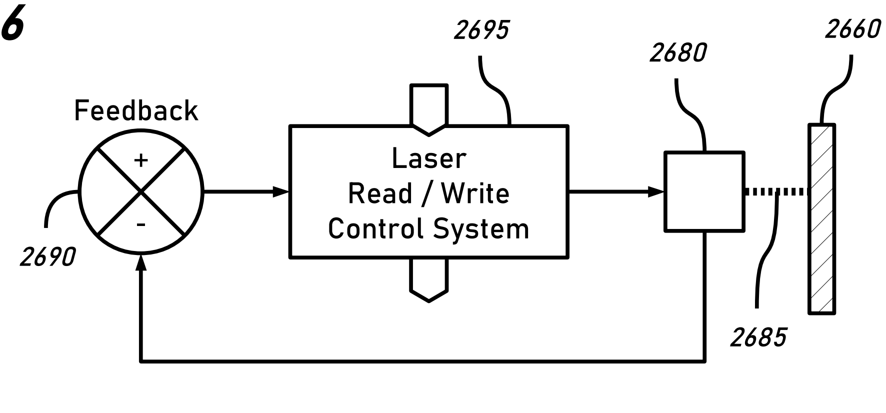
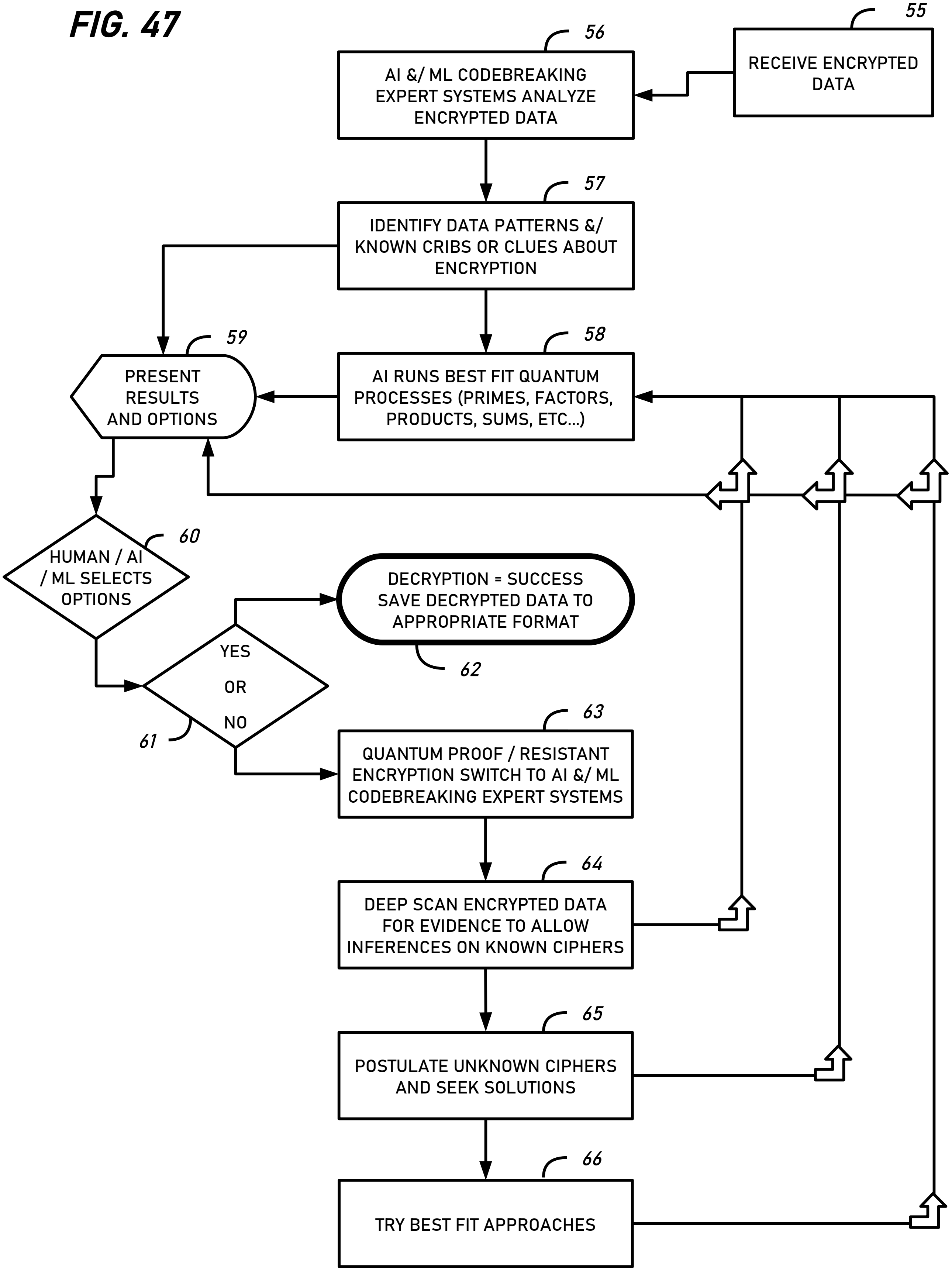


FIG. 47


55
RECEIVE ENCRYPTED DATA
56
AI &/ ML CODEBREAKING EXPERT SYSTEMS ANALYZE ENCRYPTED DATA
57
IDENTIFY DATA PATTERNS &/ KNOWN CRIBS OR CLUES ABOUT ENCRYPTION
58
AI RUNS BEST FIT QUANTUM PROCESSES (PRIMES, FACTORS, PRODUCTS, SUMS, ETC...)
59
PRESENT RESULTS AND OPTIONS
60
HUMAN / AI / ML SELECTS OPTIONS
61
YES
OR
NO
62
DECRYPTION = SUCCESS SAVE DECRYPTED DATA TO APPROPRIATE FORMAT
63
QUANTUM PROOF / RESISTANT ENCRYPTION SWITCH TO AI &/ ML CODEBREAKING EXPERT SYSTEMS
64
DEEP SCAN ENCRYPTED DATA FOR EVIDENCE TO ALLOW INFERENCES ON KNOWN CIPHERS
65
POSTULATE UNKNOWN CIPHERS AND SEEK SOLUTIONS
66
TRY BEST FIT APPROACHES

DUAL ISOLATED CHANNELS ENCRYPTION

CROSS REFERENCE TO RELATED APPLICATIONS

[001] This application claims benefit under 35 U.S.C. § 119(e) of priority to U.S. Patent Applications:

(A) Ser. No. US 17151086 – filed January 15, 2021, (issued as) US 11956352 B2,

(B) which claimed priority from provisional Ser. No. US 62961228, filed January 15, 2020,

(C) and priority to pending U.S. application No. US 18439687, filed February 12, 2024,

(D) and priority to pending U.S. application No. US 17339935, filed June 4, 2021, which are hereby incorporated by reference in their entirety.

FIELD

[002] The subject disclosure relates to data security, data protection in storage, data protection in transit, encryption, ciphers, secure communications, interception mitigation, and decryption. It furthermore relates to artificial intelligence, machine learning, quantum computing, and information processing systems. Civilian, military and intelligence applications of these subject technologies are included.

BACKGROUND

[003] As the world's premier producer of high value intellectual property, and advanced defense technologies the United States has suffered and is still suffering from industrial scale intellectual property theft. Mainly but by no means exclusively by China. Plus, the menace of Russian hacking and ransomware attacks.

[004] Industrial and economic espionage is a high-value, low-risk crime that many nation-states and some corporations indulge in. Yet the chances of being prosecuted for intellectual property theft are exceedingly low. Even though the value of some intellectual property can be exceptionally high. Consequently, for a high-tech criminal, it is a safer and better paying crime than most others. For which very few people are ever caught and even fewer are prosecuted.

[005] The military needs more secure information systems, networks, and communications, right down to the level of reducing signal noise / leakage from systems such as missiles, tanks, and aircraft. Because this noise may be captured by signals intelligence (Sig-Int) eavesdropping. Signal noise and leakage can allow eavesdroppers to reconstruct the information on a display screen from nothing more than signal leakage. This is why the U.S. prohibits its employees who may possess sensitive information from using a personal computer in hotels within the Peoples Republic of China.

[006] Intelligence agencies seek to steal military secrets and intellectual property. Organized criminals steal intellectual property and other commercially valuable information for economic gain. There is considerable crossover between state and criminal intellectual property theft as well as more generalized theft of data.

[007] There is also a black market for stolen credit and debit cards on the Dark Web. Much economic crime is comprised of the theft of financial information.

[008] Furthermore, there is also an increasing risk from perverts who may wish to steal images and illegally access the rising number of devices with cameras to spy on and photograph children or people they are unlawfully stalking. This storm has not yet broken in the media.

[009] Mobile devices are perhaps one of the greatest security threats posed to most organizations, and to individuals and their privacy. This is because they are carried with us and can yield large amounts of personal data. Particularly offensive dangers are posed by pedophiles accessing the phones of children, to track and stalk them, and to take pictures of them and to groom, train and control them. Yet this has to be balanced against the upside which includes that many parents find it comforting and helpful to be able to track, and coordinate with their children using mobile phones.

[0010] A very large part of our modern lives is now documented, organized and stored online in live systems. Banks, retailers, governments, and the military have huge amounts of data housed online. The U.S. Military has its own version of the Internet, with its own data centers and cloud infrastructure.

[0011] In Amazon's and eBay's favor, online shopping is a "*green*" success story. Providing a less polluting solution than consumers driving to and from shopping malls. Unfortunately, e-commerce systems and their data are extremely attractive targets. Which can and do suffer large-scale data theft. Unfortunately, online services tend to store large amounts of data in one place, and in standardized formats. Data which is just sat on servers long-term such as the entirety of user data possessed by Facebook, Wells Fargo, or Google creating a target akin to a big "*Buffalo*" made out of data. Presenting a big stationary target.

[0012] Hackers try hard to break into these big stationary targets. Whether it is perverts trying to get at children's data on Instagram, or financial crime, or espionage – the systems that are often most in need of protection are the back-end systems. Those which hold the databases, containing user accounts and personal information. Plus, databases for email, chat, and passwords that are often held locally on user devices.

[0013] Securing streaming media is important for the protection of intellectual property rights; and so that content can be metered and paid for. The motion-picture industry is a valuable national asset. When piracy injures that industry, it hurts the North American economy. The problems of streaming media and public Wi-Fi use are in some ways similar to some of the problems faced by the military.

[0014] Protection against interception and decryption is of vital importance for the military and the intelligence community. Because loose lips really may sink ships. Military data in transit needs robust, fault tolerant systems, that are also secure.

[0015] Military applications are more demanding, because they have to protect the data created and needed by war fighters, in real time, and in life-or-death situations. So, both Hollywood and the U.S. military can benefit from improved protection of data streaming technology; and users of public Wi-Fi may benefit from improvements to device-to-device security.

[0016] The main protective technology in the prior art is the mathematics based cryptographic encryption of data. For which many encryption programs that are usually based on prime numbers and the factors of numbers have been created. Ciphers and codes have a long history which began with spying and the passing of secret messages.

[0017] As information technology evolves, new risks emerge with each new addition to the technology. So too the defenses against abuse must evolve. Unfortunately, most if not all encryption that is based on clever mathematics can be hacked by hostile intelligence agencies.

Furthermore, just as humans have now lost the battle for superiority in chess to computers. We will probably soon will lose it in other areas to Artificial Intelligence and Quantum computing systems. That are able to process multiple possible solutions in parallel. The coming soon Quantum computing technology has been heralded as the end for mathematics-based encryption. Which has been dubbed the "*Quantum Apocalypse*". We don't know for sure how long it will be before Quantum computing renders current encryption technologies obsolete. This patent is made in anticipation that the end of encryption as we have known it in this Information Age may be with us by around 2030.

[0018] The subject technologies are intended to be part of the solutions which are needed already for military and intelligence communications. The military and intelligence community need to quantum proof their communications and data storage right now or as soon as possible. Because some secrets need to stay secret for a very long time. Plus, strategic rivals will be warehousing as much data as they can collect now; and aiming to decrypt it once Quantum Decryption is sufficiently operationalized. Perhaps in combination with artificial intelligence.

SUMMARY

[0019] In one aspect of the disclosure, an encryption system for digital communications is provided. The system includes a first device coupled to a telecommunications network through a first network connection. The first device is configured to operate on a first channel to transmit and receive data. The system also includes a second device coupled to the telecommunications network through a second network connection. The second device is configured to operate on the first channel to transmit and receive data. A third network connection is configured to connect the first device to a second channel in the telecommunications network. A fourth network connection is configured to connect the second device to the second channel in the telecommunications network. The first channel is isolated and separate from the second channel. Encrypted data passes between the first device and the second device through the first channel. Decryption data configured to decrypt the encrypted data in the first channel, passes between the first device and the second device through the second channel.

[0020] In another aspect of the disclosure a method of encrypting digital communications is provided. The method includes establishing a first channel to transmit and receive data through a first network connection of a first device coupled to a telecommunications network. A second device is connected to the first channel through a second network connection coupled to the telecommunications network. A second channel is established in the telecommunications network. The first device is connected to the telecommunications network via a third network connection. A fourth network connection is established and configured to connect the second device to the second channel in the telecommunications network. The first channel is isolated and separate from the second channel. Encrypted data passes between the first device and the second device through the first channel. Decryption data configured to decrypt the encrypted data in the first channel, passes between the first device and the second device through the second channel.

[0021] In another aspect of the disclosure, a computer program product for encrypting digital communications is provided. The computer program product comprises a computer readable storage medium having program instructions embodied therewith. An execution of the program instructions cause a processor to establish a first channel to transmit and receive data through a first network connection of a first device coupled to a telecommunications network. A second device is connected to the first channel through a second network connection coupled to the telecommunications network. A second channel is established in the telecommunications network. The first device is connected to the telecommunications network via a third network connection. A fourth network connection is established and configured to connect the second device to the second channel in the telecommunications network. The first channel is isolated and separate from the second channel. Encrypted data passes between the first device and the second device through the first channel. Decryption data configured to decrypt the encrypted data in the first channel, passes between the first device and the second device through the second channel

[0022] Consequently, it should be understood that many other possible configurations and combinations of the subject technology will become readily apparent to those skilled in the art from this specification generally and the following detailed description, wherein various configurations of the subject technology are shown and described by way of illustration. As will be realized, the subject technology is capable of other and different configurations or combinations and its several details are capable of modification in various other respects, all without departing from the scope of the subject technology. Accordingly, the drawings and detailed description are to be regarded as illustrative in nature and not as restrictive.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] Figure 1 is a mixed schematic and systems block, flow diagram of two systems communicating according to various aspects of the subject technologies.

[0024] Figure 1A is block flow diagram of two systems operating two communication channels according to an aspect of the subject technologies.

[0025] Figure 2 is a mixed schematic and systems block, flow diagram of two systems communicating according to various aspects of the subject technologies.

[0026] Figure 3 is a mixed schematic and systems block, flow diagram of two systems communicating according to various aspects of the subject technologies.

[0027] Figure 4 is a block flow diagram of defined process according to an aspect of the subject technologies.

[0028] Figure 5 is a block flow diagram of a cipher creation and distribution process according to aspects of the subject technologies.

[0029] Figure 6 is a bar chart graphing a statistical analysis of a text sample in accord with an aspect of the subject technologies.

[0030] Figure 6A is a bar chart graphing a statistical analysis of a text sample which has been statistically adjusted in accord with an aspect of the subject technologies.

[0031] Figure 7 is a defined process for cipher creation and distribution according to aspects of the subject technologies.

[0032] Figure 8 is a schematic representation of characters from a text sample processed according to aspects of the subject technologies.

[0033] Figure 8A is a schematic representation of characters from a text sample processed according to aspects of the subject technologies.

[0034] Figure 9 is a block flow diagram of a quantum secured channel according to an aspect of the subject technologies.

[0035] Figure 9A is a mixed schematic and block flow diagram of an aspect of a quantum secured channel according to an aspect of the subject technologies.

[0036] Figure 10 is a mixed schematic and block flow diagram of an aspect of a quantum secured channel according to an aspect of the subject technologies.

[0037] Figure 11 is a block flow diagram of a quantum secured network or harness according to an aspect of the subject technologies.

[0038] Figure 11A is a mixed schematic and block flow diagram of an aspect of a quantum secured channel according to an aspect of the subject technologies.

[0039] Figure 12 is a mixed schematic, and systems block flow diagram of two systems cooperating according to various aspects of the subject technologies.

[0040] Figure 13 is a mixed schematic, and systems block flow diagram of two systems cooperating according to various aspects of the subject technologies.

[0041] Figure 13A is a block flow system diagram of an aspect of the subject technology used as a black box module according to the subject technologies.

[0042] Figure 14 is a mixed schematic, and systems block flow diagram of two systems cooperating according to various aspects of the subject technologies.

[0043] Figure 15 is a mixed schematic, and systems block flow diagram of two systems cooperating according to various aspects of the subject technologies.

[0044] Figure 16 is a schematic diagram of an array of character values and their index configured according to an aspect of the subject technologies.

[0045] Figure 17 is a schematic diagram of an array of character values, string values and more complex data object values according to an aspect of the subject technologies.

[0046] Figure 18 is a block and flow diagram defined process according to an aspect of the subject technologies.

[0047] Figure 18A is a block and flow diagram defined process according to an aspect of the subject technologies.

[0048] Figure 19 is a block flow diagram of cooperating local and remote systems processing data according to aspects of the subject technologies.

[0049] Figure 19A is a block flow diagram of cooperating local and remote systems processing data according to aspects of the subject technologies.

[0050] Figure 19B is a block flow diagram of cooperating local and remote systems processing data according to aspects of the subject technologies.

[0051] Figure 20 is a hierarchical block diagram of aspects of the subject technologies.

[0052] Figure 21 is a block diagram relating to the prior art.

[0053] Figure 22 is block and flow system diagram of aspects of the subject technologies.

[0054] Figure 23 is a block and flow systems diagram of the security of alternate file processes according to various alternate aspects of the subject technologies.

[0055] Figure 24 is block and flow diagram of file processing according to an aspect of the subject technologies.

[0056] Figure 25 is a schematic drawing of file processing according to an aspect of the subject technologies.

[0057] Figure 26 is a schematic drawing of file processing according to an aspect of the subject technologies.

[0058] Figure 27 is a schematic drawing of file processing according to an aspect of the subject technologies.

[0059] Figure 28 is a block and flow systems diagram of the security of alternate file processes according to various alternate aspects of the subject technologies.

[0060] Figure 29 is a block and flow diagram describing a logic module, for writing data blocs to random storage locations according to an aspect of the subject technologies.

[0061] Figure 30 is a block and flow diagram describing a logic module, for writing data blocs to random storage locations according to an aspect of the subject technologies.

[0062] Figure 31 is a block and flow system diagram of the reconstruction of a randomized file according to an aspect of the subject technologies.

[0063] Figure 32 is a block system and flow diagram of the reassembly of randomized data according to various aspects of the subject technologies.

[0064] Figure 33 shows randomized data being retrieved from parallel storage locations according to an aspect of the subject technologies.

[0065] Figure 34 shows a modified file being separated into new data blocks written randomly, according to an aspect of the subject technologies.

[0066] Figure 35 is schematic plus a logic module in block and flow system diagram form, performing a secure data deletion according to an aspect of the subject technologies.

[0067] Figure 36 is an image being broken apart by a pattern cut into data blocks according to an aspect of the subject technologies.

[0068] Figure 37 is an image being broken apart by a cut pattern into data blocks according to an aspect of the subject technologies.

[0069] Figure 38 is an image being broken apart by a cut pattern into data blocks according to an aspect of the subject technologies.

[0070] Figure 39 is a mixed schematic block and flow system diagram of the creation of a stream of data blocks created from an image, according to an aspect of the subject technologies.

[0071] Figure 40 is a mixed schematic and block and flow system diagram of the logic and creation of data blocks from a motion picture, according to an aspect of the subject technologies.

[0072] Figure 41 is a mixed schematic and block flow system diagram of secure cloud storage and uploading data in accord with various aspects of the subject technologies.

[0073] Figure 41A is a mixed schematic and block flow system diagram of secure cloud storage and uploading data in accord with various aspects of the subject technologies.

[0074] Figure 42 is a mixed schematic and block flow system diagram of secure cloud storage and downloading data in accord with various aspects of the subject technologies.

[0075] Figure 42A is a mixed schematic and block flow system diagram of secure cloud storage and downloading data in accord with various aspects of the subject technologies.

[0076] Figure 43 is a mixed schematic and block flow system diagram of secured streaming downloads in accord with various aspects of the subject technologies.

[0077] Figure 44 is a drawing of a bank card according to the prior art

[0078] Figure 45 is a credit card according to an aspect of the subject technologies.

[0079] Figure 46 is a laser read and write system operated according to an aspect of the subject technologies.

[0080] Figure 47 is a block and flow system diagram of a binary artificial intelligence and quantum decryption process according to an aspect of the subject technologies.

DETAILED DESCRIPTION

[0081] The detailed description set forth below is intended as a description of various configurations and / or combinations of the subject technology and is not intended to represent the only possible configurations and / or combinations in which the subject technology may be practiced. The appended drawings are incorporated herein and constitute a part of the detailed description. The detailed description includes specific details for the purpose of providing a thorough understanding of the subject technology. However, it will be apparent to those skilled in the art that the subject technology may be practiced without these specific details. Like or similar components may be labeled with identical element numbers for ease of understanding.

[0082] In general, embodiments of the subject technology improve upon the state of the art, and / or their applications. Alternative variant embodiments can also improve upon the state of the art in the systems into which they are incorporated, and / or their applications. In the drawings this symbol "&" means "and", this symbol "/" means "or" and this combination "&/" means "and / or".

[0083] These subject technologies are intended to provide a different approach, from the purely mathematics-based encryption and decryption technologies in the art. The problem with all the currently used encryption-based security measures is that they all rely heavily on abstract mathematics. Which is embodied in programs that transmute the original data into data which can only be decrypted by reversing the abstract mathematical steps made in the creation of the encrypted data. Computers can relentlessly keep trying until they crack such known encryption types. This may be the main weakness of the technology in the state of the art. Because what one talented mathematician can encrypt, an equally talented mathematician can decrypt. Powerful modern computers increase the speed and complexity of the calculations. Furthermore, quantum computers that can process multiple possibilities at once and in parallel are probably going to render most of the encryption in the art obsolete within the next five to ten years. Which process may be further accelerated by hybridizing quantum computers and artificial intelligence.

[0084] The subject technologies seek to take some of the abstract math out the equation, save that some math can be used to disguise the fact that this technology does not rely purely on mathematics per se. Though it is logical, and thus amenable to computer processing and control.

[0085] These subject technologies are intended to create an offset to these coming challenges to data security and secure communications. Randomization may be introduced and combined with logic and programs that operate the methods and systems of these subject technologies. As well as creating and attributing meaning to data structured according to logical protocols.

[0086] The military work in contested environments including a contested electromagnetic spectrum. They have to communicate secret time sensitive orders, strategic and targeting data reliably and securely. Even when adversaries are intercepting and decrypting or jamming those communications the military has to be able to communicate securely and reliably. There are commonalities between these military needs and the needs of civilians.

[0087] The efficacy of many forms of encryption including in the current art may be boosted by the use of an analytics adjusting cipher (AAC) prior to another encryption. AAC performs statistical adjustments of the populations of characters and / or data pattern frequencies. As explained in Figure 5, Figure 6, Figure 6A, Figure 7, Figure 8, and Figure 8A. Data streaming aspects are more particularly addressed in relation to Figure 42, Figure 42A and Figure 43 along with image processing and other aspects of randomization from Figure 37 to Figure 40. The Data Security Using Randomized Features (DSURF) subject technologies are explained in relation to the drawings from Figure 19 on through to Figure 43. Which provide quantum-proof and hacking-resistant cloud data center security. As well as providing a data streaming capability (Figure 43), and portability from a DSURF secured cloud storage, to another cloud services provider and / or to local devices. Where a local version of DSURF may also be operated to provide similar functionality on user devices.

[0088] As explained above, the encryption used on the Internet by most users most of the time is not very secure. It may fend off some criminals but not all, and it will certainly not stop professional spies. So, the Inventor has created a new transient type of cipher-logic and programs, for use between two nodes which may be two battlefield computers or radios, or two civilian devices such as phones, or computing tablets. This aspect of the subject technologies is Time Randomizing Interface Protocol Language Encryption (TRIPLE) of US 11956352 B2 and US 18439687.

[0089] In which two devices may use a randomizing interface protocol to create a unique language known as a "*transient random interface protocol language*" (TRIPL) that may be used for a limited and randomizable time period. After which it may be replaced by another TRIPL. An AAC may be applied before TRIPLE, which has the effect of improving the efficacy. Akin to switching up through 128k to 256k or 512k etc. math-based encryption standards. Furthermore, an AAC may be used prior to a 128k or 256k or 512k math-based encryption to boost their efficacy similarly to how an AAC may be used to boost the efficacy of TRIPLE.

[0090] The time period of TRIPL transience may also be randomized and varied sufficiently often to minimize the benefit to be gained from cracking and deciphering any one iteration. To provide an interface language that is governed by randomized protocols according to these subject technologies.

[0091] TRIPLE may provide a solution to battlefield and local encryption needs, and also for encryption used across wider networks. TRIPLE creates a unique randomized cipher that may exist between communicating points or nodes on a network that may be used only between those communicating nodes and may be used by them only fleetingly. Before being re-set to a newly created TRIPL. A current TRIPL cipher version may be used during the creation of the next TRIPL and so onward. By adjustment of the timing of replacement of the data patterns or characters of a TRIPL, the characters or data patterns that make up that TRIPL may also be individually replaced on a rolling basis. Providing a TRIPL that is constantly transient and constantly evolving to the next TRIPL while in use. So that once two nodes are operating securely using TRIPLE, it should be very difficult to break into those communications in an efficient and timely manner or at all. Data traveling over networks may travel through many TRIPLE ciphers.

[0092] So that intercepting those communications over time that include use of many unique TRIPL none of which are likely to ever be recreated, and which may last only minutes, or even seconds before being replaced. Each TRIPLE interface being programmed to randomly negotiate a new TRIPL at randomly timed intervals, and to operate like a "*black-box*". Which allows devices on the inside of the TRIPLE interfaces to work just like regular devices plugged into black-box communications like a bespoke router that only they can communicate with.

[0093] TRIPLE may thus create a constantly moving target, that it is intended not to be worthwhile hacking.

[0094] TRIPLE can also be used between any two devices with the appropriate software according to these subject technologies. This software may also be provided in firmware for use within networking cards and routers and similarly embedded into systems and hardware. So that TRIPLE may be used for all device to device, or peer to peer communications.

[0095] Furthermore, as between two devices one unique TRIPL may be used, and a different TRIPLE cipher may be created at every interface between nodes as information routes through a network. The same data may thus pass through one unique TRIPLE for every pair of nodes passed through, and within each such interface the language / cipher may be unique, and time restricted. There is no reason why more than two devices could not share the TRIPL if system designers wished to configure it that way. Which may be more attractive to designed when TRIPLE is used over a DICE protected communications link.

[0096] The creation of some TRIPL for use in TRIPLE communications are addressed in the descriptive materials pertaining to Figure 12, Figure 13, Figure 14, Figure 15, Figure 16, Figure 17, Figure 18 and Figure 18A, and in those figures themselves.

[0097] This explanation continues with simple examples for aid of clarity. Within these systems a character such as "H" in the ASCII may be represented by a binary code or other base code.

[0098] With no encryption this "H" or its binary code can be sent over a network, and codebreakers may hack this data feed to intercept it by various means. However, in normal circumstances a codebreaker would probably also intercept the character that preceded and followed the "H" if there were any such preceding or following letters.

[0099] Data stored or transmitted according to the DSURF subject technology which is also based on randomization but is not based on timing or transience; provides that data is broken apart into blocks, and where the individual data blocks go to be stored is randomized. So that, for example, Cloud based systems may have been reconfigured each with its' own number or network address. For example, there could be one thousand individual servers within the cloud data center (simplified down for reason of space in the drawings to five possible storage locations). Each with its own network address.

[00100] The data may be stored into this cloud by dividing it into data blocks (the size of which may or may not be randomized). Then sending each data block to a randomized location within one of the one thousand servers, and the locations to which the data was sent is recorded as a series of pointers: within an array to make a sequential key comprising the location in order of storage of every data block. Which key may further be comprised of sequentially stored pointers.

[00101] The storage of which key may be located within another separate specialist location that may for additional security be kept separate and isolated from the data storage locations. This key may alternatively be stored on a local hard drive or backed up to USB drives or other media perhaps in a hidden drive, or hidden file capable of acting like an access dongle. The possibilities are endless, and some of these options are more secure than others. These examples illustrate some but not all possibilities and are not intended to limit or confine the scope of the subject technology.

[00102] This key and the randomized storage allocation, as provided within DSURF should not be confused with "*Hashing Functions*". Which are mainly about efficiency not security per se. Furthermore, while it is true that a hashing function could be based on randomization, and the programs of the subject technology might conceivably be used as part of a hashing function. Due to a modest overlap of suitability, hashing functions are about mathematical efficiency employed for optimization of storage times and access times for data retrieval. Randomization of data storage may provide one efficient method. But these subject technologies are designed to prevent data collisions, and to prevent damage to preexisting data. Whereas hashing functions achieve efficient storage by mathematically avoiding rather than preventing data storage collisions. Consequently, they may sometimes accidentally overwrite data due to collisions and then have to reinstate it or hold two sets of values mapped to the same location. Hashing is mainly about writing and reading data quickly, not about data protection and security.

[00103] Careful study of these subject technologies and the two decision tree programs of Figure 29 and Figure 30 contain solutions that are designed as computer program logic modules capable of preventing potential data write collisions and / or data overwrites. Potential data write operations that would be overwrites or collisions may be detected and prevented before any data is written. Because the objectives and guiding rationale are about data protection and security in these subject technologies.

[00104] So that in these subject technologies according to Figure 29 any data in a location selected randomly for writing new data may be logically tested before writing 500, and if a potential for collision with existing data is indicated – then another data write location may be generated 510 to write the data-block; or where a storage location is available but too small to accommodate the data block, another storage location may be produced 500. Alternatively, according to Figure 30 a data-block partial-write – plus an overflow operation 565 may be produced 569. So that the surplus data overflows and is written to another storage location 570. In which case the first location includes a pointer to the overflow location 575; or the pointers-key includes the overflow location.

[00105] The basis of selection of data-block storage locations is based on randomization, and data protection. Rather than the potential for efficiency of data writes or access and read times per se.

[00106] This fact, however, does not need to preclude the use of the most efficient possible program to achieve the subject technologies. Which may or may not be compatible with any given hashing (efficiency, per se) function. Furthermore, once data is randomized while it might seem illogical, there is no particular problem with subsequently moving it around and storing it according to any particular hashing function or other management system. Provided that the original pointers key can be updated or augmented to record any changes made.

[00107] These subject technologies need not get in the way of efficiency at the macro-level, or micro-level. The subject technologies just should not be confused with such optimization methods and / or "*hashing*" functions per se, or similar procedures, nor confused with fragmentation or defragmentation in the context of drive optimization. They are specific things. These subject technologies can be made compatible with such other things, but care must be taken. Because disk management programs can damage data processed according to these subject technologies. This need not be a problem, because a terminate and stay resident (TSR) program may be left running in the background on a computer system to modify its behavior to one compatible with these subject technologies. It may be helpful to think of hashing functions primarily aimed at speeding-up data access times for read and write operations. They usually map two dimensional constructs called tables.

[00108] Fragmentation is also usually expressed to represent drive storage spaces as two-dimensional constructs or tables. This is a natural side effect of the way some drive management systems operate, and the way they are typically processed. There are drive management systems that can defragment fragmented data to enable the disk space to be used more efficiently.

[00109] These subject technologies are not primarily aimed at either speed of access per se; nor are they aimed at disk optimization, fragmentation, or defragmentation per se. Indeed, these subject technologies can be set to write very dense unfragmented, even compressed randomized data where slack space may be utilized for storing randomized data. This has an additional benefit in that the more files are randomly intermingled, the more they may each protect the other.

[00110] The two modules of Figures 29 and 30 may be used to achieve dense data-block storage, at or close to the contiguous maximum storage capacity of a storage space, partition, or drive. Though this is not a specific goal pursued in these subject technologies, developers may nevertheless find it useful to be aware of this aspect. The amount of available space may be used to determine the most appropriate data writing method.

[00111] Programmers have designed hashing functions, fragmentation, and defragmentation around the notion of tables and two-dimensional maps. Computers do not need to visualize data in ways that are human compatible. In these subject technologies, the key may be comprised of a one-dimensional array for ease of use and simplicity. Computers, and especially artificial intelligence, having no problem conceiving one very long one-dimensional structure. Like a theoretically infinitely extendable street, with specific data at every address.

[00112] For example, data is stored randomly into storage, the locations of which are then recorded in a long one-dimensional array. Coordinates in two dimensions are often not necessary. Additional computation and cross referencing of variables in more complex human friendly structures such as tables, using some sort of coordinate system for read/write data referencing may thus be avoided.

[00113] The ethos of these subject technologies is that they are intended to be minimalist, practical, clean, and as simple and hence as efficient as possible. With the focus on data protection from corruption and security threats.

[00114] The pointers key may be comprised of a simple array containing the locations recorded in the sequential order over time, of where each block of data was sent for example. To retrieve the data from the cloud, the cloud computers are requested to return whatever was stored at each specific location identified by the pointers key for reassembly back into their original order. A good way to visualize this as shown in Figures 41 through 43 is rather like a zipper, which clicks data blocks apart for storage, and zips the same data blocks back together to resemble them back into their original order.

[00115] It may be appreciated that this process could be virtualized and / or scaled down to run on just one device such as the internal storage of a smartphone, or PC drive, and it could still provide a serviceable data protection and security capability. Which in the absence of the key would be close to impossible to reconstruct. Similarly, the subject technology may run on just a suitably programmed CPU working with only RAM or flash memory, provided there is a suitably reliable non-transient data storage capability.

[00116] Major targets for hackers are servers holding financial and user data. Presently this is relatively easy for them. Because the files are in one place and will even have helpful end markers between files. How nice of the systems designers to make our files so easy to find and to add bookends to parse each file.

[00117] Whereas if hackers were to steal the entire contents of the one thousand cloud servers of a data center configured according to the DSURF subject technologies – there is no presently known decryption program which could reassemble any user or account data that has been stored according to these subject technologies. It would just be a mountain of useless gibberish without the key comprised of the sequential location pointers, and comprehension of what it is, relative to the data blocks to which the pointers correlate.

[00118] Possession of one user's key of pointers may only identify that user's data within the system, and keys may be handled on a separate channel to data and stored into a separate location. Whereby keys and data may be kept isolated from each other. If stolen a pointers-based key cannot be used to identify the data of any other user. So that for a hack to succeed in getting mass user data, it could only work if all the keys relating to all users of it were also stolen. Thus, making the technical challenge of bulk data theft from DSURF protected systems potentially insurmountable.

[00119] Users operating a version of DSURF on local devices may also store their keys in locations, none of which is co-located, and instead is isolated from their data. Furthermore, a behemoth like AWS might have a thousand data centers. Each with a thousand servers and be able to randomize storage over a truly awesome amount of cloud locations. There is strength in numbers in IT, as there is in nature, for herding animals such as Wilder-Beast that are confronted by a predators. Which is based on leveraging the mathematics of scale.

[00120] Suffice to say that the bigger the total number of files stored as data blocks into random locations, within available storage – the harder to crack they all become. When the pointers key is also kept online similarly and isolated away from the data according to these subject technologies, perhaps at a different location. Then the user experience may be similar in terms of performance to that possible in the less secure current state of the art.

[00121] When the user device pulls both keys and data down from the cloud only as needed as shown in Figure 42, Figure 42A and Figure 43. Then together they can provide a seamless and fast access to user data that is similar to the present level of user experience. But with improved data protection and security.

[00122] Because the only way to reassemble one user’s data is to access the correct locations and to reassemble the data back into the correct order regardless of where they are stored. The Cloud data itself may thus be rendered useless to hackers who are looking to steal user data in bulk. Because it is probably going to be impossible for any hacker, however gifted, to ever know or discover the randomized order and / or locations into which any specific user's data was stored.

[00123] The storage record on the user’s device, USB drive, or cloud key storage provider is the only key capable of recovering the data. Furthermore, even if hackers managed to get the key of one user, it would be of no help to them whatsoever with the rest of the data in the Cloud infrastructure. So that it is hoped that all the big data handling corporations such as Facebook, AWS, Google, Apple, and Microsoft will want to use the Cloud based version of the DSURF subject technologies to keep their user data safe. Certainly, the Inventor hopes to license it to all of them as well as to the Department of Defense (DoD).

[00124] Vitally though, there are also benefits to be gained at the macro level for data holding organizations including Banks and retailers. Who can also reap benefits from these subject technologies. Because even if one user is careless with his or her data key or dongle, it will give no clue nor any way to access anything else. With the miniaturization of flash memory and chips onto bank cards, these may also be made capable of holding enough data to be used to store randomization keys comprised of arrays of pointers and / or unique historical user or tally data on credit cards as shown in Figure 44 and Figure 45. The DSURF subject technologies may be capable of preventing many large-scale data thefts.

[00125] Whenever users are operating a device which is accessing the Internet it must do so via various protocols and data interface standards etc., one of which is that the device must have an Internet Protocol (IP) address. Mostly IP addresses are provided to end users by Internet Service Providers (ISP) and mostly while the ISP may in its Terms and Conditions promise to deliver a floating changeable IP address; in reality most of the time end users get the same IP address. Furthermore, end users can request that they be given a fixed unchanging IP address. Anyone who hosts their own website, for example, will need a fixed IP address. Whereas most Internet users do not need a fixed IP address.

[00126] IP addresses map to geographic locations. When users visit websites the website operator can look up their IP address to discern their approximate location. This seems to have escaped the attention of many banks and financial institutions who are concerned primarily with establishing the identity of their users. However, this approach is creeping into use for intellectual property licensing and streaming media. Probably because these are already carved up according to geographic and jurisdictional rules.

[00127] However, these measures can sometimes be circumvented by using proxies, and / or Virtual Private Networks (VPN) that have geographically distant proxies in another country for example. IP addressing can thus be used in addition to the randomization subject technology, to add another layer of verification-security.

[00128] In a transaction according to this aspect of the subject technologies tally data such as the IP address plus date and time stamps of the last time the banking website was accessed can be recorded on the device such as a PC, Smart Phone or Tablet or bank card. Furthermore, an entire history of previous access data, comprising IP addresses and dates can be recorded on devices and bank records. Historical data can be saved and used as a virtual "*tally-stick*" or comparator which should be identical at both ends.

[00129] However, when a user accesses their bank account from a new device, the bank may compare the data of the new device. A failure to tally can thus flag either that the legitimate client is using a new device, or that a fraudster is attempting to access the account. So that where the tally data does not match, further checks may be triggered to ensure that the current user is the true account owner and not a fraudster.

[00130] In another case a new device may replace an old device, so the IP addresses may be similar or identical, but no tally data exists on the new device. Again, extra challenge questions may be used. After which a new tally file can be written to the new device. Tally data may be ported to a new device from an old device.

[00131] Challenge questions may include one such as: "*Is this the first time you have used this device to access your account?*" Further challenge questions may be added to ensure the user's ID is verified and tied to the new device. Of more concern to financial and business institutions may be when the IP address does not tally with previous IP and geographic data. Again, when the IP address and / or the system time zone running on a device tracks to hacking hot spots in Russia, China or South America as opposed to the user's home country. Then this sort of discrepancy may be used to raise a red flag; and trigger extra security checks.

[00132] Thus, by use of the IP addresses and the tallying of historical interactions recorded as between a user's device and the networked service; service providers may screen out hackers using stolen log-in data; but who cannot provide tally data of IP address usage, and event logging histories from previous logins – nor answer challenge questions. Thus, suspicious transactions may be declined, and the incidence of theft may be further reduced.

[00133] Artificial Intelligence programs operating fraud pattern recognition methods may also be used to identify variations in user behavior patterns and may raise red flags accordingly. This may be accomplished by recognizing activity as being outside of expected parameters.

[00134] In the case of streaming media one or more data blocks can take the form of a "*stripe*" of data written to or read from a specific cloud storage location. The locations of which stripes can switch among a population of servers. On which the streaming data is being or has been randomized into storage according to DSURF.

[00135] A sequential reading of a stripe of pointers 1505 as that shown in Figure 43 can be used to access and buffer the streaming of media such as cinematographic works, or live events to consumer viewing devices. An appropriate timing window can be used to buffer the streaming data and streaming pointer keys needed to call it from the host locations may provide, like a time sensitive key.

[00136] Users need not be aware of this complexity of DSURF as pointers keys may be downloaded in parallel also using separate threads with stripes or data blocks. Data threads may stream through different ports as systems designers may prefer. Indeed, port usage variants offer a further possibility for randomization as well as for channel isolation.

[00137] These aspects of the subject technologies may prove very useful and well suited to secure pay per view entertainment, and for keeping communications secure. Because for security applications the randomized stripes of randomized data blocks can be switched not only between server locations but also between channels in the case of secure voice and video communications, or any communications.

[00138] In the case of secure networks such channel switching may be introduced along with the server switching. So that even live communications data may be caused to random channel switch, random route-switch, and random server or location within server switch according to the steam of pointers. A stream of pointers may also be provided through channel switching, route-switching, and server or location within server switching. Reading data ahead of its due time into a buffer may be used to smooth the process out for users.

[00139] In secure communications as the military may need, channel-hopping radio communications might also be used to provide the stream of pointers for other streaming or live communications. Channel-hopping requires both nodes to "*hop*" or switch channels according to a shared program. Which is sufficiently unpredictable (or pseudo-random) to prevent eavesdroppers from being able to correctly guess the next channel hop. For the users of channel-hopping communications they may be relatively unawares of, and hence unimpeded by the fact their devices are continuously switching channels.

[00140] The possibilities are endless for routing the streams of pointers differently than the streams of data blocks to provide isolation between them. So that interception of either one will not easily yield useful time sensitive information to an adversary. So that switching data channels according to a preset but secret program, as well as these subject technologies randomizing attributes of the data itself, can be leveraged. Thereby to make it very difficult for an adversary to get enough data blocks or stripes to be able to resemble them, even where some channels are intercepted some of the time. These subject technologies may be used in the modern battle space for networked communications.

[00141] Time shifting stripes of data may also be used to move data blocks such as a frame of audio-visual data relative to other frames within a sequence. The frame may be randomly shifted out of sequence; and the pointers needed to shift a data stream back into the correct order to play correctly used as a means to scramble and unscramble data.

[00142] Communications may also benefit from randomization in real-time. Here is a simple example. The interfaces of two systems may begin to communicate with each other for the first time. Their programming tells them to agree on the meaning of a set of characters to create a simple transient randomizing interface protocol language (TRIPL) for use in encryption. They both generate a random number between maxima and minima. They exchange their initial numbers add them together and divide them by two to agree the common number rounded-up to the nearest integer represents an “A”; then they repeat the process until they have completed the alphabet, and all the numbers needed for a number base system and any other symbols that may be desired. Thereafter they have a common set of characters in a language that only they know.

[00143] They may use this language for a random time period. Then once one or other randomly timed between maxima and minima triggers a reset, they may create a new randomized language for use over another random time period, or they may periodically agree new values for characters individually also on a rolling basis.

[00144] Voices can be converted to text, and text can be converted to speech. So synthetic speech can be communicated as text using a TRIPLE protected communication channel. This may be of considerable utility for battlefield communications and for sensitive telephone calls to be made by intelligence operatives who are in hostile locations. Via an application that converts the speech to text, applies TRIPLE, and disguises their voices in calls back to their headquarters for example. So those agents cannot be identified by their voices, and their conversations can be secure.

[00145] More complex implementations can be made to communicate more complex data such as images and video as well. Even without complex implementations binary data which is very simple can be converted to bigger bases agreed similarly randomly. Then the numbers represented by randomly chosen numbers that correlate to the base in which the numbers are being expressed.

[00146] It is not possible to list all the possibilities. These are just a few examples at the simple end of what is possible using these subject technologies. Having introduced various aspects, in the context of a few examples of the subject technologies and an outline of which drawings are most applicable to them – more specific aspects are now explained in more depth in relation to the drawings.

[00147] Similar terminologies: transient random interface protocol language (TRIPL) and time randomizing interface protocol language encryption (TRIPLE) may have similar and overlapping meanings. TRIPLE is achieved, using a TRIPL. So, all TRIPLE includes use of a TRIPL, the time randomizing aspect of TRIPLE is where one TRIPL is replaced by another, in the broader TRIPLE process. Timing replacements may be randomized on a rolling basis for characters of the TRIPL once it has been created and / or for the whole TRIPL. TRIPLE systems and methods may be used by information processing machines, and humans may in some cases operate under a TRIPLE system or method and create TRIPL. The efficacy of encryption, including TRIPLE may be increased by using an AAC to adjust the statistical frequencies of characters; so as to make it harder for codebreakers to infer or guess the values of characters or patterns in the underlying data based on statistical analysis.

[00148] Aspects of Dual Isolated Channels Encryption (DICE) systems, methods, and procedures may be used in combination to boost existing forms of encryption in the art. Such as ones based on mathematics that have a public and private key, and ones that have only a private key that is shared. Ones that have only a shared private key and no public key would be ideally suited for use with DICE. Where the two channels may carry the private keys for each other but not on their own data channel where those keys are used.

[00149] Though the materials of this specification mainly describe DICE as it may be used with randomization and logical cipher-based encryption technologies such as DSURF, AAC and TRIPLE. This is a similar situation to the example where two isolated channels may be used to provide DICE with purely mathematics-based encryptions from the prior-art.

[00150] DSURF begins with the embodiment shown in Figure 19, the similar embodiment of Figure 19A, and the embodiment provided in Figure 19B then is built out from there into randomized cloud data protection and security through the figures which go into detail leading up to sub-systems of Figure 41, plus Figure 42 which deal with two similar uploads, then in the sub-systems of Figure 42 and black-box process 1497 of Figure 42A which deal with two similar downloads, to those shown operating in Figure 43. Which deals with streaming data using pointers-based keys – to call for downloads of data blocks or stripes as data streams. Downloading from random storage locations stored according to a system operating DSURF.

[00151] Figure 1 shows DICE being operated between two devices. The subject technologies of TRIPLE, AAC, and DSURF may all be used with DICE which may serve to boost the efficacy of their security. This is achieved by operating two separate channels that may be isolated from each other per se. So, to be clear TRIPLE, AAC, and DSURF do not have to be used with DICE, nor does DICE have to be used with them. But their efficacy may be improved by it.

Similarly to the way that other forms of encryption based purely on mathematics in the art that have only private keys may have their efficacy improved by operating them with and according to an implementation of DICE. Furthermore, DICE and those other subject technologies TRIPLE, AAC, and DSURF may be bootstrapped from systems operating the encryptions in the art such as from a web-browser running HTTPS, including the private key and public key pair-based encryptions. Using any of the subject technologies may boost the efficacy of any encryption and may be compatible with most if not all of the encryption technologies in the state of the art as of 2025.

[00152] In Figure 1 there is a first device “A” that is a smart phone 5, which is connected to a second device “B” that is also a smart phone 8. First device 5 and second device 8 are connected through their cellular networks. First device, phone “A” connects through cellular network “C” 6, and second device phone “B” connects through cellular network “D” 7, and the two cellular networks “C” 6 and “D” 7 connect with each other. These connections provide the cellular networks’ ability to connect many phones to many phones, and in this case, they are used to provide the first data channel 1 that may be operated as a quantum decryption resistant or quantum decryption proof data channel 1 that may securely carry user communications data 11. Both devices (5 and 8) may use their cellular network connections to exchange data back and forth in communications with each other and with others as peers,(such as websites and their cloud computing or generative artificial intelligence provider). All such communications may be configured and operated similarly to this example.

[00153] Phone device 5 “A” also has a software application (or “App”) based “e-SIM” card or virtual SIM card 3 installed and running on it. The app may run separately to the physical SIM card and operate separately with its own cellular network “E” 9 providing the phone device 5 “A” with a second channel 2. As we will see next, the apparatus and method used to gain this second channel capability does not matter per se. The phone device “B” 8 achieves a second channel capability via, for example, a coupled Wi-Fi router 4, which may be connected to a fiber optical network “O” 10. The network 10 in turn is connected with cellular network “E” 9. This series of connections provides a second channel 2 that is separate from the first channel 1 and may be isolated from the first channel 1 and vice versa according to the subject technologies (as shown in detail in Figure 1A) does not depend on any particular configuration of hardware. The isolation of the channels 1 and 2 from each other may be created by not bridging the two channels with each other within the devices. So that each channel may send and receive data. But neither channel can connect directly with the other. Details of encryption and / or encryption keys and / or protocols 12 may be stored into a location from one channel, then retrieved from that location, and used to operate the encryption of the data 11 on the other channel and vice versa without there being a direct way for those channels to interact /or connect to each other. This is an illustrative example of many possible variants that is not intended to restrict the subject technologies.

[00154] The second channel 2 may then operate between the two phones to provide cipher or encryption key negotiation and / or key distribution 12. Through which details of the encryption to be used on the first channel 1, to encrypt the data 11, may be negotiated and / or shared according to the TRIPLE and / or AAC and / or DSURF subject technologies. Furthermore, this second channel 2 may also be used to similarly according to the DICE subject technologies to share details of any desired encryption to be used on the first channel – including math-based encryptions, ideally based on secret keys using DICE for private key distribution / or sharing 12.

[00155] The first channel 1 between phones “A” and “B” is incapable of operating any known system of quantum protection using currently available cellular telecommunications infrastructure. Because it is based on transmitting and receiving radio waves in both directions. So that this first data channel 1 needs and may using DICE be provided with quantum decryption resistant or quantum decryption proof encryption. But may not use any form of quantum encryption end to end that is based on entangled photons. Though it is possible that parts of any communications that travel over the Internet may travel over fiber optical networks that may have the capability to use quantum effects-based protection. That may apply over some parts of the channel.

[00156] The second channel 2 between the Wi-Fi router 4, and through the Optical Network "O" 10 may be partially quantum encrypted and / or protected from the Wi-Fi router through the Optical Network 10 which may extend over the land based optical Internet through optical fiber cabling to connect to the cellular network "E". Which in turn completes the last stage of its connection to phone "A" via the e-SIM 3, that is also using radio-wave-based cellular network technology, that is incompatible in the state of the art with quantum encrypted and / or protection based on quantum-effects such as entanglement.

[00157] The second channel 2 may be protected using any suitable encryption including quantum entanglement-based protection at stages that operate over optical networks. Indeed, to avoid confusing these examples the switching of the roles of the two channels is entirely possible, and furthermore the first channel 1 may be used as the data channel for data traveling through it and may double as an encryption / key channel during a different stage of communication provided that the keys and the data they protect are always kept on different channels that are isolated from each other. Which arrangements may thus be symmetrically configured on both channels to boost speeds and efficacy in both directions. An illustration of symmetrical communications using DICE on both channels are provided in Figure 4A, and Figure 4B.

[00158] The devices may thus share and / or negotiate encryption for the first channel 1 that may be a quantum decryption proof encryption using the separate channel 2. This may use AAC and / or TRIPLE and / or other forms of encryption including the math-based encryption in the art, preferably the ones that have only private keys. Which thus provides Dual Isolated Channels (14) Encryption.

[00159] Persons skilled in the art will appreciate that DICE may be expanded to use a larger plurality of parallel channels and that data may be sent in parallel through them in stripes and switching over both channels or a larger plurality of channels in order to further improve data security in transit, and to improve data transfer rates. Because the bandwidth and data transfer rates achievable using DICE may be equal to the sum of the bandwidth and data transfer rates of such a plurality of channels. However, for many users and applications, this is unlikely to be warranted. But for military and intelligence applications and even within networks a larger plurality of channels operating based on the DICE systems, processes, and methods may be used to leverage the power of secure parallel processing. Which may be especially useful in the fields of graphics processing and artificial intelligence which may be accelerated using parallel processing. Which parallel processing capabilities could be further improved securely by using DICE's parallel communications and increased bandwidth potential, as well as using it for improved security. For the sake of clarity, one might think of those potential super-scaled variants as Multiple Isolated Channels Encryption (MICE). However, for the purposes of this specification it is for the sake of clarity that the subject technology is explained in terms of two channels. Two channels is the minimum number of channels, there being no upper limit to the number of channels that may be used in parallel and according to the DICE subject technologies.

[00160] The greater the physical separation of the channels the more likely it is that greater security is achieved. For example, probably the least secure embodiment of DICE would probably be based on operating at least two separate channels through the technique known as operating “threads”, and / or using different ports as a means of separating channels, but which may share the same physical infrastructure. This is not to say that operating DICE according to such an embodiment would not improve security over that available in the art, because it would. But it may have a weakness against collection of data 11 and encryption keys 12 from just one physical infrastructure at some points during transit of the communications data between sender and receiver. So that it is unlikely to deliver the full benefits of the DICE subject technologies. Nevertheless, it may still provide a low-cost embodiment of the subject technologies that can provide some of the increased security benefits of DICE and is intended to be included within the scope of these subject technologies. Figure 1A shows the core of the operation of DICE which is intended to be compatible with all possible embodiments including ones based on using at least two threads, or two ports to provide the separation of at least two channels.

[00161] Figure 1A shows the mechanism at the heart of the isolation of the first channel 1 and second channel 2. There is a symmetry as between the two communicating devices at the functional level in terms of both the apparatus and method used to operate them. There is also another symmetry between the first channel 1, and the second channel 2. So that describing the operation of the channels operated by either device is identical to the other. The DICE application 50, is loaded into working RAM 51 and is running on the CPU 49. From where the DICE application operates the input and output (I/O) 52 for the first channel 1, and the input and output (I/O) 53 for the second channel 2.

[00162] A DICE application may provide write operations from the second channel 2 into internal storage 32, at the bespoke channel 2 read / write storage system (directory) 34. The DICE application may provide read operations from the internal storage 32, at the bespoke storage system (directory) 34 to provide details of encryptions, encryption keys (12) and any applicable protocols to an encryption application that may perform encryption operations on communications data passing through the first channel 1.

[00163] A DICE application may provide write operations from the first channel 1 into internal storage 32, at the bespoke channel 1 read / write storage system (directory) 33. The DICE application may provide read operations from the internal storage 32, at the bespoke storage system (directory) 33 to provide details of encryptions, encryption keys (12) and any applicable protocols to an encryption application that may perform encryption operations on communications data passing through the second channel 2.

[00164] There is no connection or coupling across the first channel 1, and second channel 2 other than as described above. The channels are thus isolated from each other 14 and do not connect to each other. They have the ability to write data into bespoke storage locations applicable to each other but not for themselves. DICE reads that data from those bespoke locations and passes that data to the encryption application to be applied to the opposite / or another different channel from the bespoke storage on the channel from which it was read. Available bandwidth across these channels may be summed to provide an overall total bandwidth if they are operated symmetrically and in parallel. Details are provided in the description pertaining to Figure 4, Figure 4A and Figure 4B.

[00165] Figure 2 shows a directory containing files that have been encrypted and secured using DSURF 91. That is being sent in combination with DICE using two parallel channels. There are at least two possible ways this may be accomplished. Figure 2 provides just one half of the picture, which is compatible with explaining both of those. Which would be identical and symmetrical except for the data and keys being on the opposite channels to the ones shown which is not shown for this example. Furthermore, the symmetrical mode of operation is not essential to the DICE subject technologies. So that this figure could equally be the entire and full presentation of a DICE operation that is simple and is not symmetrical. In which the directory of files encrypted by DSURF may be sent through the first channel 1, and the pointers-based key used in the DSURF technology (explained below) is sent through the second channel 2. The drawing may represent that scenario, as well as representing half of a symmetrical DICE implementation in which multiple encryptions may be running. With data and applicable keys always separated onto opposite channels according to the DICE subject technologies.

[00166] Providing an illustrative example of a directory containing files that has been encrypted using DSURF 91 and sent through the first channel 1 from phone device “B” 8, to phone device “A” 5. Then with isolation between unbridged channels 14, and the DSURF key 12 sent the second channel 2 from phone device “B” 8, to phone device “A” 5. Where phone device “A” may use the key 12 to unlock and reconstitute in decrypted form the opened files 94. Again, this example could be run the other way around, and the roles of the channels switched. Provided that the data and the encryption keys remain isolated from each other on separate and unbridged channels, then interception of either, without the other will be of no use to bad actors. In symmetrical operation, both channels may be used to provide quantum decryption proof encryption to each other. That has total bandwidth comprised of the sum of the bandwidths of the two channels.

[00167] The NSA reportedly does not in the ordinary course of events collect the actual communications passing over networks per se in America. But they may be able to see both channels connecting the two phones. They could probably in most cases using databases and cross-referencing identify the two channels being used for their purposes. But within America it is less likely that bad actors, including bad state actors would be able to collect the traffic of both data and keys. Frankly, if a hostile spy agency wants to know what someone is saying on their phone, they have many other and better ways to do it. Such as installing key loggers and using the phone's microphone to bug a target. Which may be enabled via "*Trojan Horse*" and "*Spear*" attacks that do not necessitate interception of communications per se in most cases. They can bug a target's home, or clothing for example. So that while DICE is not going to stop the American government agencies doing their jobs, it should be good enough to stop bad actors from hacking their communications. DSURF provides an excellent solution to protecting stored data and using DICE to send DSURF protected data over public network infrastructures can make the sending of large files a lot more secure.

[00168] Figure 3 provides another example of messaging using DICE to protect a message such as an email or text message 95. Wherein the message may be encrypted by AAC and / or TRIPLE and / or other encryptions or ciphers and which travels as data 11 through the first channel 1 from phone device "B" 8 to phone device "A" 5. Which first channel 1 may be isolated from the second channel 2 by the use of separate unbridged channels 14 (as explained above in relation to Figure 1A) within and between the devices "A" 5, and "B" 8. Wherein details of the encryptions AAC and / or TRIPLE and / or DSURF and / or other math-based encryptions in the art and their secret keys 12 travel through the second channel 2. To phone device "A" 5, where the encryption keys 12 are applied to decrypt and open the message 94 the data of which has traveled over the first channel 1.

[00169] Again, for the sake of clarity the symmetrical view showing potential dual roles of both channels, is not shown. But if it were shown the drawing would be almost identical except that keys 12 would be passing through the first channel 1, and data 11 to which those keys apply would be passing over the second channel.

[00170] Figure 4 shows a predefined process that provides as follows:

"69. The local system and remote system may begin using any encryption such as HTTPS via web-browser or no encryption, operating through a telecommunications network infrastructure and / or the Internet.

70. Initiate contact with remote system through Channel 1.

71. Request dual isolated channels encryption.

72. If accepted exchange encryption keys, plus any rules and protocols:

> *73. create and write to / or retrieve local system private key from*
>
> *Channel 2 storage (34)*
>
> *74. initiate connection to remote system through Channel 2*
>
> *75. send private key through Channel 2.*
>
> *76. Start send and / or receive encrypted communications data exchange over Channel 1 using key(s) and any protocols stored in Channel 2 storage (34).*
>
> *77. Receive / or provide remote system private key through Channel 1:*
>
> > *78. write private key to Channel 1 storage (33)*
> >
> > *79. retrieve private key from Channel 1 storage (33).*

80. Start send and / or receive encrypted communications data exchange over Channel 2 using key(s) and any protocols stored in Channel 1 storage (33).

81. Periodically refresh / generate and exchange new keys according to a parameter (that may be randomly timed or set to zero to switch off).

82. Retain keys in storage upon instances of dropped or lost connections.

83. Resume using stored keys or if corrupted loop back (→ 70).

84. Terminate at end of session – may include preset option to retain keys in storage (33, 34).

85. Resume using stored keys or if corrupted loop back (→ 70).)"

[00171] Figure 4A shows a mobile phone device "V" 54 that is operating a first channel 1, and a second channel 2 according to the DICE subject technologies having executed all the steps of the defined process 54 down to and including step 83 using both channels symmetrically 103. Whereby it is sending data 11 in both directions over both channels, limited only by their bandwidth carrying capacity. The sequence of letters on the data blocks or stripes 11 indicates a sequence. Indicating that the data once broken into stripes or blocks may also be sent in an alternating sequence across the channels. So that the maximum speed of the two channels may be achieved.

[00172] This brings two very important benefits. Improved speed of communications, rendering DICE systems capable of providing enhanced speeds in both directions. Plus, improved data security because the blocks or stripes of data on each channel are (i) separated, (ii) isolated from each other, and (iii) encrypted using different encryption keys. So that anyone who intercepts channel 1, even if they managed to decrypt that data 11 they would get only "A", "C", "E", "G", "I", "K", "M", and "O" from channel 1. Which may have an almost unbreakable encryption, and even if it could be decrypted would constitute only half of the data 11. Because the other data blocks or stripes of the file or stream "B", "D", "F", "H", "J", "L", "N", and "P" were sent on channel 2. Which is unlikely to be identified per se, as well as being also subjected to a strong encryption with a different key. So that there are many difficult challenges to anyone who seeks to collect and decrypt the data 11 sent using DICE in this example.

[00173] Figure 4B shows a personal computer device “W” 67, that may be the device that the phone device “V” 45 of the previous figure may be communicating with. Using DICE symmetrically 103 to send and receive the data 11. Using a personal computer with DICE may become highly desirable for online gaming, graphics processing, and artificial intelligence applications. Not just for the increased security but because those applications can benefit from the improved speeds provided by operating DICE channels symmetrically. Which ideally may become the default for many users. Seeking to get the most out of their gaming experience and maximize their ability to leverage the increased speeds for artificial intelligence and graphics heavy applications.

[00174] Figure 4C shows how a user may integrate DICE into his or her online activities using their mobile phone “V” 54, laptop computer “W”, and router 90. In ways that would seem natural to most people who use mobile phones and computers and who have access to their home internet services through a router. Indeed, most people in a Starbucks coffee house will have access to all the physical apparatus needed. With the possible exception of having an e-SIM card 96 or virtual SIM-card application installed on their mobile phone. These may be installed on any smartphone and DICE may be used to operate them in combination with a physical SIM-card. Similarly, most people will have access to a Cloud computing infrastructure 135 for data storage, and / or for their generative artificial intelligence activities. That are accessed over the Internet through a radio-wave-based connection 99 to a cellular network “X” 92 using their physical SIM-card, and / or through a radio-wave-based connection 98 to a cellular network “Y” 93 using a virtual SIM-card application, and / or to a cellular network. Alternatively, the phone device “V” 54 may also operate DICE by connecting to and through the Internet via a connection 67 to a personal computer “W” that may be sharing its connection to the Internet 89 through a router 90;

or the phone “V” 54 may also connect to and through the Internet via a connection 68 to and through a router 90.

[00175] The personal computer in this scenario “W” 67, may connect to and through the Internet via a connection 67, to the mobile phone “V” 54 which may share its connectivity with the computer to provide a channel for use with DICE in combination with the computer “W” 67 using channel 89 connected to and through the router 90. So that for anyone who owns a mobile phone, with a contract with a cellular network provider may use DICE, and anyone who also owns a computer with networking capabilities may using their phone and router to provide two separate and isolated channels may also leverage the extra security and speed of DICE for their communications with other devices that are also operating DICE on their devices. Which may include providers of Internet and Cloud computing based services.

[00176] The router 90 merits further consideration, because in the future Internet service providers such as Verizon and others may wish to provide routers that have two fiber-optical channels so that they may be able to provide a highly secure and high-speed version of DICE implemented into their routers and as part of their services for users. They could do this at least two ways. By either multiplexing two separate channels through one fiber optical cable (most likely) to a next node in their infrastructure which then may operate to separate the two channels to ensure their data 11 may travel separately over different infrastructures or have time-shifted separations from there, or by using a pair of fiber optical cables from the router. So that the router 90 may provide a plurality of channels 97, through its optical network “Z” 91 and on to online resources 101 and services in Cloud infrastructure 135. So that DICE may be utilized using various combinations of the available channels available to persons who have a mobile phone, a personal computer, and access to an Internet connected router. Which may be in their homes, and / or at workplace, and / or restaurants and bars. Persons skilled in the art will appreciate that it would be equally possible to operate DICE using a satellite-based network such as Starlink to obtain Internet connectivity. They will also appreciate that there are other possible configurations of apparatus and connectivity between devices capable of operating according to the DICE technology without escaping the scope of these subject technologies. Regardless of whether or not devices are communicating peer to peer, or with online services such as generative artificial intelligence or Cloud storage or online shopping or other services. All of which may benefit from the improved security and speeds that DICE may provide.

[00177] Figure 5 shows a flow diagram for the process of creating an Analytics Adjusting Cipher (AAC). AAC may be capable of assisting or providing quantum decryption resistant and / or quantum decryption proof encryption. The AAC works by removing the cribs and clues that codebreakers use. The mathematics used is based on statistical analysis and adjustments to the data set, comprising the data to be encrypted. Which adjustments are compiled into a cipher, that may be recorded into a sequential array or table that may be indexable per se. For example, in most messages the most common character that occurs is the “ ” space character, which below the ASCII character code is based on a data pattern of binary ones and zeros. A codebreaker who understands this would probably try to find all the space characters as a first step towards decryption. This is because spaces go between and parse every word. So, from this one character may be deduced the length of each word in a message. This is why the AAC randomly disguises space characters which are substituted by a variety of wildcards under certain rules. As is explained in the worked example from a sample of text, which is analyzed and processed according to the method of AAC. Details of which are provided in the descriptive materials pertaining to Figure 5, Figure 6, Figure 6A, Figure 7, Figure 8, and Figure 8A in combination with those figures.

[00178] The first step of creating an AAC is to receive a sample of text 15 or other communications data. This data sample may be in a particular language 16, and of a type of communication of the same or similar type to that to be encrypted. The total populations of characters and / or data patterns as well as the total sum of all characters are calculated 17. Then comes the calculation of coefficients or fractions of occurrence of the populations (P) of characters or data patterns 19, *P/ΣP* 20. Which results may then be presented along with options 18. So that the user who may be human or an artificial intelligence is able to make those decisions 21 and make selections accordingly within an interface. As a result of user decisions, the user and / or an artificial intelligence and / or machine learning module may apply policies 23. That provides instructions to create randomized character and / or data pattern replacement ciphers 22 by substitutions using wildcards. So as to randomly apply substitute specific wildcards for members of specific populations to reduce population totals from above target parameters. Which may be the mean or average or median frequency of occurrence of the totals of the populations 24. Then record those analytics adjustment ciphers or keys that map to arrays and records 25.

[00179] This takes care of all the populations that were above the target parameter, and an example of the mapping of those substitutions into arrays and records is set out in Figure 8 and Figure 8A. Figure 6 shows the analysis of the text sample before adjustment 3155. Figure 6A shows analysis of the same text after the adjustment of all those populations that were above the parameter after they have been statistically adjusted down to the target parameter 3156. It’s very visually clear between the two graphs that the space character and other more prevalent character populations no longer stand out so much.

[00180] If so desired as well as adjusting populations down to a parameter, smaller populations can be adjusted up to a parameter. By adding in disinformation characters as wildcards

26. In the worked-example, this optional alternative step has not been used. Users and system designers may tweak the AAC process and method to their own parameters and requirements without deviating from the subject technologies. The ACC and / or its addressable pointers mapping keys may then be shared or distributed to connected devices 27. The sharing of the AAC pointers mapping keys to an array containing the AAC 28, may use a connection that is a separate (unbridged) and an isolated parallel channel according to the DICE subject technologies for sharing AAC and / or other keys that may also be secured by an encryption 29. So that the connected and communicating devices may apply the AAC (alone or in combination another encryption) to data files and / or communications 30, as well as applying any other encryptions in combination with the AAC 31.

[00181] The graph of Figure 6 relates to analysis of the following sample of text:-

> *"I was a student in England, with two children Tommy and Holly. It was summer vacation, and we had gone on vacation with only a hundred pounds, a tent, and a full tank of gas.*
>
> *We went to the seaside town Skegness. The first campsite we tried turfed us off; we had to take down our tent and look for somewhere else to camp. We were all tired, and it was dark. So, I decided we would just have to fly pitch anywhere we could.*
>
> *I drove to the coast where we found a bay, with a wide grass verge. On our radio the DJ said there was a meteor shower that very night. I decided to fly-pitch in the bay, atop the cliffs, with an amazing view out to sea. Thankfully, Tommy loved building tents, and so he did most of the work. While Holly and I organized some food and drinks. I had a four pack of beer. We set up our moonlit supper then sat back and watched the meteor shower. With clear skies, looking out to sea where there was very little light pollution. The meteor shower was the most intense I've ever seen. Music played on the car radio. I explained all about asteroids, meteors and the extinction of the Dinosaurs, and the kids watched the meteor shower with awe and amazement. It was one of those*

> *perfect moments in life. When time seems to stand still for a fleeting moment. While all was right with the world, my kids and I were safe and happy together. Then Holly said some words I will never forget, and treasure always: "Dad, you're not like other people's dads or anyone else I know. Other parents are boring, they would never do anything like this. This is really cool!!!" My heart filled with pride and love – in what was one of the happiest moments my children and me ever shared. On the night when we were so poor, we could not afford to stay in a regular hotel or resort or go on a regular vacation like other families."*

[00182] The sample text was subjected to statistical analysis of the total numbers of specific characters. Organizations that use a specific nomenclature may vary somewhat in their use of language. This statistical analysis may be usefully tabulated, as in the table below: The left most first column shows the character, the second column shows the total number of occurrences "*n*" for each population (P), and the third column shows the number of occurrences of characters as a decimal fraction of the population. These figures are used to derive a probability-based coefficient for each character which values are listed in the fourth column. These coefficients are rounded to one integer value in the fifth column, rounded to one decimal place in the sixth column, and rounded to two decimal places in the seventh column.

Char	Number *"n"* Occurrences	*n / population*	Probability Coefficient	Coefficient Integer	Coefficient 1 Decimal	Coefficient 2 Decimal
a	113	0.062122	1.9879054426	**2**	2.0	1.99
b	7	0.003848	0.1231445849	**0**	0.1	0.12
c	25	0.013744	0.4398020891	**0**	0.4	0.44
d	78	0.042881	1.3721825179	**1**	1.4	1.37
e	187	0.102804	3.2897196262	**3**	3.3	3.29
f	30	0.016493	0.5277625069	**1**	0.5	0.53
g	22	0.012095	0.3870258384	**0**	0.4	0.39
h	78	0.042881	1.3721825179	**1**	1.4	1.37
i	92	0.050577	1.6184716877	**2**	1.6	1.62
j	2	0.001100	0.0351841671	**0**	0.0	0.04
k	18	0.009896	0.3166575041	**0**	0.3	0.32
l	66	0.036284	1.1610775151	**1**	1.2	1.16
m	36	0.019791	0.6333150082	**1**	0.6	0.63
n	90	0.049478	1.5832875206	**2**	1.6	1.58
o	121	0.066520	2.1286421111	**2**	2.1	2.13
p	23	0.012644	0.4046179219	**0**	0.4	0.40
q	0	0.000000	0.0000000000	**0**	0.0	0.00
r	83	0.045629	1.4601429357	**1**	1.5	1.46
s	83	0.045629	1.4601429357	**1**	1.5	1.46
t	130	0.071468	2.2869708631	**2**	2.3	2.29
u	31	0.017042	0.5453545904	**1**	0.5	0.55
v	16	0.008796	0.2814733370	**0**	0.3	0.28
w	61	0.033535	1.0731170973	**1**	1.1	1.07
x	2	0.001100	0.0351841671	**0**	0.0	0.04
y	26	0.014294	0.4573941726	**0**	0.5	0.46
z	3	0.001649	0.0527762507	**0**	0.1	0.05
.	24	0.013194	0.4222100055	**0**	0.4	0.42
,	19	0.010445	0.3342495877	**0**	0.3	0.33
;	1	0.000550	0.0175920836	**0**	0.0	0.02
'	3	0.001649	0.0527762507	**0**	0.1	0.05
!	3	0.001649	0.0527762507	**0**	0.1	0.05
	346	0.190214	6.0868609126	**6**	6.1	6.09
Totals	**1819**	**1**	**32**			

[00183] Capitalization, and Tab-Shift or paragraph indentations have been ignored, and it has been assumed that potential codebreakers know the language of the sample is English. These steps were taken to keep this example simple and for clarity. According to the AAC method of statistical adjustment chosen, those characters which occurred at twice the average rate and above according to the integer values of the fifth column of the table were then selected as likely targets. Which most codebreakers would try to identify first. As a way to find a crib, or clue and way to begin to break any code or cipher applied to the text sample.

[00184] Probably a codebreaker would first try to find a repeating data pattern, probably as a binary representation of the space character, because not only is this six times more likely to occur than any other character. If found as a repeating pattern in the data, it could be used to parse the words of the text sample. Word lengths might then be inferred. This is why the space character “ ” is the single most dangerous character in the sample. Disguising it makes sense as part of any AAC.

[00185] Codebreakers will also know the approximate statistical metrics for text in English even though this text is from the Inventor who has an English accent and style of writing that has taken on some American influences due to living in America. So, it makes sense to also reduce the average frequencies of occurrence of other characters that occur at an above average rate. In this case those with a probability coefficient of two or more according to column five, which has been rounded to integer values. This step identified all the characters with an integer coefficient of “2” or above as “a” equals 2, “e” equals 3, “i” equals 2, “n” equals 2, “o” equals 2, “t” equals 2 plus the space character which equals 6 shown in those columns, and which are indicated 3160 in Figure 6.

[00186] The vertical axis of Figure 6 shows the number of occurrences “*n*” as the total number of occurrences of each of the characters indicated on the horizontal axis “a” through “z” plus punctuation. These data are shown as vertical bars 3160 in the bar chart.

[00187] The AAC method is to identify the set of most vulnerable characters 3160 that code-breakers will try to use, then to statistically adjust them to blend into the background of less useful data. This is accomplished by adding wild card substitute characters to the set of characters 3160, as shown in Figure 6. Figure 6A, indicates where the characters 3160 have been randomly reduced to a range of maxima and minima about the average frequency of occurrence by re-distributing their total among the original character values 3160 plus however many wildcards for “a” 3161, “e” 3162, “i” 3163, “n” 3164, “o” 3165, “t” 3166, and “ ” 3167 characters are needed in each case.

[00188] So that of the total occurrences of value “a” these are randomly reduced close to the average value by randomly substituting approximately half of them with “ϟ”; “e” substituted by approximately one third “﷼” and one third “Ψ” and so on down the values culminating in the space character “ ” at the bottom of the column of values and wildcards, which had five sixths of their total population (P) randomly substituted within maxima and minima totals with the wildcard 3161 characters “И”, “Ӂ”, “گ”, “۞”, “‡”.

[00189] It should be apparent to persons skilled in the art that wildcard characters should be drawn from the same seed data set as any TRIPL they are to be used with. No TRIPL character or data pattern should be used to represent more than one actual character from the original data unless a bespoke AAC reverse cipher or look-up capability is to be used along with it. Generally, while it may be possible without departing from these subject technologies in some cases it is not a preferred approach in this specification.

[00190] Figure 8A continues this process by also substituting the remaining non-wildcard characters of the adjusted populations 3160. Which may be part of an AAC process or as for TRIPL character values so that “a” will be substituted for by “⫨” 3170, and so on throughout the values in the columns, leading to an AAC and / or TRIPL cipher that has been optimized by the substitution of the AAC wildcards 3161. Now in this figure the AAC wildcards 3161 have been carried through to the final cipher. However, this has been done mainly for illustrative purposes. In practice programmers who may implement the AAC method to suit their own needs may prefer to run a TRIPL creation process on the entire data set after the AAC wildcards have been added. It makes no difference to the relative statistical effects either way. But in this illustration, it is hoped that it is clearer to see the method at work by not substituting (again) the wildcards in the final right column values 3180. This aspect makes the most sense when Figure 8 and Figure 8A are studied. Because the effect is clear, and results in the cipher substitutions shown in their arrays.

[00191] In the case where an AAC may be used to make other forms of encryption and ciphers harder to crack, then it may be used either with TIPLE or without it as a stand-alone security booster for otherwise encrypted information. It having the benefit that text processing such as is deployed within AAC and TRIPLE are not demanding on computer processor resources. Unlike the higher end mathematical encryption which uses intentionally difficult and demanding computation as the main barrier to decryption.

[00192] Figure 6 shows on the vertical axis the total number of occurrences “*n*” of each population (P) of characters, against the characters themselves on the horizontal axis for characters “a” through “z” plus punctuation, in the pre-AAC analysis of the text sample. The populations of characters that need to be adjusted may be visually identified from the tallest bars on the bar chart 3160 and from their corresponding values in the data analytics table (above).

[00193] Figure 6A is the same on the vertical axis but the AAC adjusted totals for “a” plus wildcard substitutes 3161, “e” plus wildcard substitutes 3162, “i” plus wildcard substitutes 3163, "n" plus wildcard substitutes 3164, "o" plus wildcard substitutes 3165, "t" plus wildcard substitutes 3166, " " plus wildcard substitutes 3167, are much lower. The visual impact of the AAC is very clear in this bar chart form. It is possible to visualize how much the ability to find useful cribs and data patterns has been reduced by the application of an AAC to the text sample. On the horizontal axis the wildcards can be seen to the right of the character they have been substituted for in this statistically adjusted sample which shows the peak values are now all within the chosen parameter.

[00194] Figure 7 provides a predefined process for the creation of an AAC 35. Persons skilled in the art may understand that even where the relative statistics have become well known through use on a certain type of communications data or documentation – that a new sample of communications data need not be received on every run through this process. But that at some point the key metrics of a sample of, or sample compatible with the applicable data must have been obtained from some sample of data at some time. The defined process steps provided 35 are as follows:

"36. Receive data sample.

37. Count total identifiable data objects or characters.

38. Identify every type of data object or character.

39. Count totals for every data object or character.

40. Calculate the mean, median, and / or average values for each data object or character.

41. Select adjustment parameters:

42. adjust by random substitutions with wildcards the items with populations of above a parameter down to within that parameter (which may be a percentage of or within a deviation from the mean or median or average population size), *43. (optional) adjust by random additions of (disinformation) wildcards the to items with populations of below a parameter (which may be a percentage of or within a deviation from the mean or median or average population size),*

44. (optional) substitute the remaining original data objects and / or characters within a population using wildcards.

45. (optional alternative to step 44) apply another cipher such as Time Randomizing Interface Protocol Language Encryption (TRIPLE) and / or any other compatible cipher and / or mathematical encryption.

46. AAC and other ciphers may be shared with correspondents, and / or negotiated with correspondents via a separate channel that is isolated from the channel that is used to carry the data to be encrypted by the cipher(s).

47. The isolated cipher and / or key channel (if used as such) may be further protected by encryption which may be a strong encryption such as quantum encryption, and / or quantum-anti-interception measures.

48. Optionally, a purely mathematical encryption may be applied to the encrypted data. (Where AAC and / or TRIPLE ciphers are run within a webpage or application that may be running under HTTPS, and / or using other encryption in the state of the art as of 2025, or the future state of the art during the life of this patent.)

[00195] Returning to analysis of the data, the beneficial effects of the AAC process are clearly visible in Figure 6A. Where it can be seen that the space characters, which were the code-breaker's best hope for an easy way to parse words and crack the code, have been randomly substituted by wildcards so that the space character and the wildcards are all within a small deviation between maxima and minima 3167 and they no longer provide the highest single value, which is now the totals of 83 for “r” and 83 for “s”. If the codebreaker takes either “r” or “s” as being the likely crib or clue for the space character, it will lead them into chaos. Similarly, the other most commonly occurring characters will no longer provide any clue or usable crib for a codebreaker.

[00196] Persons skilled in the art will understand that in this example, characters that were at least twice as likely to occur as the average were reduced and randomized to values approximately around about the average. This could have been done with a different parameter such as the mean value rather than the average. The maxima and minima could have been varied by a different deviation from the mean or average and so on. AAC applications may be programmed with the ability to vary their own parameters, or to allow a human user or an AI user to vary them. Indeed, an integral AI module or function could be included within an AAC application computer program, to intelligently pick the best ways in which to vary these parameters for specific data sets.

[00197] So that by applying an AAC to content its attributes and underlying data can be better protected by pre-ciphering it into a state where the normal clues cannot easily be identified or used as is represented by the post AAC data analysis of the text sample, wherein the usual best hope cribs or clue characters for “a” 3161, “e” 3162, “i” 3163, “n” 3164, “o” 3165, “t” 3166, and the space “ ” 3167 character have randomized substitutions with wildcards. So that if / or when they are re-masked by a TRIPL as may be used in TRIPLE communications this will cause them to blend in and not stand out from the crowd statistically.

[00198] As they otherwise would have been easy to identify without the AAC method causing them to be distinctly randomized around the average. To give fewer clues to a codebreaker as to their true nature. Persons skilled in the art will understand the importance of the flattening and averaging of these bar graph values in Figure 6A. The adjusted totals for original characters and wildcards are shown here for clarity and continuity. Rather than the values 3180 of the next level of substitutions to remove the original characters in Figure 8A.

[00199] Figure 8 and Figure 8A have been arranged side by side, so that readers may see the substitutions of step 42 from the defined process 35 that reduced the targeted populations of characters down to within the desired parameter. As well as seeing the optional step 44 operations to substitute the remaining members of the targeted populations 3180. The arrays on the right sides of both figures are the cipher mapping arrays of the AAC, starting on the top row where original population members 3160 are partially substituted with wildcards 3161, 3162 etc. Then the remaining members of those populations 3160 such as “a” is replaced by “╧” in the AAC cipher array 3180. There are more examples listed below in the figure.

[00200] This application of AAC in this example does not presume that all characters need to be substituted. Which is because it is aimed at disrupting the ability of codebreakers to use statistical methods to find cribs with which to begin to crack the encryption to be applied after the AAC. Which may leave some characters unaffected. This is because it is likely that codebreakers will assume (wrongly) that a cipher or encryption has been equally applied to the whole of the sample text. It is unlikely they will be able to guess that a partial substitution of characters or data patterns has been intelligently applied randomly to populations of characters to manipulate statistically the relative rates of occurrence of characters into a form that is less vulnerable to the statistics-based data analytics tools of codebreakers.

[00201] This effect plus the use of a continually changing TRIPL such as that shown in Figure 14 used in TRIPLE may render these subject technologies capable of providing a very large degree of information security for the relatively low overheads associated with text processing. As opposed to the processor resource hungry higher-bit rated encryption alternatives such as 256k and 512k etc. These less processor intensive encryption and cipher technologies including TRIPLE, and AAC may not only increase the efficacy of information security, but they may also help to reduce power usage in secured systems.

[00202] In subsequent figures possible random variations are addressed. Breaking apart files into data blocks, as well as varying data block sizes according to the Data Security Using Randomized Features (DSURF) subject technologies. Which may also be made to vary randomly within desired parameters.

[00203] Figure 9 shows a quantum secured channel, that may be used for the secure transmission of encryption keys and / or ciphers and / or for data and general communications. In Figure 1, Figure 2, Figure 3, a potentially compatible channel is shown operating over a fiber optical network “O” 10 which is interposed between the Wi-Fi router 4 used by the phone “B” device 8 and extending out over the Internet from the optical network “O” 10 to the cellular network “E” 9. Which may be based upon optical fibers and optoelectronics that may be compatible with some versions of quantum-effects based security measures based on photons.

[00204] One form of quantum-effects based security uses entangled photons to secure the network against interception via wiretapping. Optical networks may be tapped by introducing a bend into the fiber optical conduit. Detection of such wiretapping is achieved because of the disruption caused to the quantum entanglement correlations of the protective entangled photon stream 451. That may be multiplexed with data carrying photons, and / or may be comprised of data carrying photons.

[00205] When the entangled photons are disrupted, they no longer correlate with their entangled counterparts. The quantum states of the retained and the transmitted entangled photon pairs 451 may be ascertained at the receiving sensors 456 which are under the control of a central processing unit 458, which operates a feedback loop 460 with the sender central processor unit 448. Which processors working together compare the entangled photons to determine if their quantum states have remained entangled. If their quantum states are correlated a match is determined 444, and if they are not correlated within a statistically significant enough number of cases 444, then it can be inferred that there is a disruption between the sender and receiver. This does not prove that there has been an interception. But it does raise a red flag, so that someone needs to go and check along the fiber optical cable. What is useful is that if the pairs are entangled in most cases to a certain level of statistical significance. Then it can be inferred that no interception or other disruption occurred over the fiber optical cabling between the sender and receiver. In this example we have not assumed that the quantum entangled pairs carry any modulated data 450. However, it may be possible to entangle photon pairs that do carry data and to use those similarly. But it is likely to be easier to use them like canaries in a coal mine to detect interceptions and disruption.

[00206] The limitations of this method of protecting communications between two nodes is not theoretical, it is technical. In that although experiments are being conducted with quantum entangled radio frequency photons, the technical challenges of using quantum entangled radio frequency photons are probably too severe for cellular phone networks to operate to an acceptable standard. This is because mobile phone signals bounce around from and go through lots of physical obstacles that would disrupt entanglements. Which is why in the examples provided in Figure 1, Figure 2, and Figure 3 neither the first channel 1, nor the second channel 2 have an end-to-end quantum encryption capability over the length of the channels that span multiple networks between the two communicating devices. This practical problem with quantum secured communications means that it is unlikely to provide a secure communications technology that is capable of use between mobile devices operating on cellular phone networks.

[00207] Ergo quantum secured communications cannot in the current state of telecommunications infrastructures provide a complete solution to quantum decryption risks. But they may help to mitigate those risks in communications over optical networks. Which is very important, because of Russian and Chinese interference with undersea fiber optical cables. Which should be detectable using quantum entangled photons 451 between nodes as shown in the quantum secured optical channel example of Figure 9.

[00208] Figure 9A shows a cipher code for letters of the alphabet being compared 444 according to a network that is operating a quantum secured channel such as that provided in Figure 9. The entangled photons may be carrying modulated data, or they may just be being multiplexed with other photons that carry the data for security. In which a sender creates a cipher 470 in cooperation with a receiver 465, and which values may be compared and used within a quantum entanglement protected communications channel 92.

[00209] This capability of quantum protected communications channels may be used within an implementation of DICE along parts of the channels created between devices. Which may work for two computers that are both connected to optical networks. But quantum-effects-based protection cannot provide an effective solution for mobile phones that connect through cellular networks. So that while these quantum technologies are helpful, a quantum decryption prevention solution such as DICE is needed. Furthermore, DICE is also needed for quantum decryption prevention over other radio wave-based telecommunications infrastructures as are operated by the military. As well as dual use networks such as Starlink, and other satellite-based telecommunications infrastructures.

[00210] Figure 10 shows various cases or outcomes for deciding whether photon pairs 451 have remained correlated or not in transit 444 after one of them has travelled through a quantum entanglement protected channel or network. In which the entangled photon source state 450 is compared with the state detected in the quantum entangled photon receiver 456. In a first case the quantum states whatever metric they are measured by is equal at both ends 461 they have remained correlated. Which determines that they are likely to be entangled. In a second case the measured metric of the quantum states does not match 462. Which may suggest an interception or other event may have disrupted the entanglement. In a third case the quantum states correlate at both ends 463 as though the photons are two halves of one shared wave function which makes it highly likely that there has been no interception or other disruption event.

[00211] Reading across all these examples it is also possible to see that a spin up photon may be used to signify either a “1” or a “0”, and that a spin down photon could be used to signify either a “1” or a “0”. Which simple attribution of meaning when operated as part of a system or method per se may be used to modulate data onto a quantum entangled photon stream 451 and to transmit it between a sender and a receiver over an optical channel. Furthermore, while there are practical difficulties with using entangled photons that are also radio waves on earth, it may be possible to use them point to point in space-based node to node communications. Which could potentially operate over significant distances.

[00212] Figure 11A is similar to Figure 9A but shows a hybrid binary and quantum harness or network that operates a binary communications channel 403. That may provide secure communications and / or quantum computation instructions to remote system parts 406 using photons 451 that may carry quantum information parameters within their quantum states. Which may be used to permit a binary part of a system to communicate with and / or through and even to operate on and / or through quantum system parts 406. As may be used to operate a quantum computer that may be connected to its various system parts over an optical network. Which may include encoding more complex quantum states onto photons 451, by modulating a characteristic of the waveform such as amplitude for example.

[00213] Such a quantum harness (implemented through a network) may have its system parts 406 distributed across a plurality of physical locations, that are connected and cooperating. That is operable over that network that is equivalent in function to a harness as may be implemented within a unitary binary and quantum computing system assembly. Where these subject technologies may intersect is where a user operates a quantum computer remotely over a quantum capable network. Which may be based on fiber optical infrastructure. That could allow quantum states to be produced in photons 451 that are then sent over a fiber optical network, provided there is an entirely optical network channel operating between the remote user device and the quantum computing networked system. Parts of which quantum computing system may be located at different physical locations provided that the quantum network can connect the various quantum system parts 406 together with each other, and with the non-quantum binary communications channel 403. The security and integrity of which communications may be checked also by comparing entangled photon pairs for correlations 444.

[00214] Figure 11A is similar to Figure 9A except for the quantum states data carrying photons 451 which instead of being made up of zeros and ones now includes the "?" to indicate that this quantum channel is sending and receiving quantum states data that are not limited to zero or one, but may represent a super-position, or other quantum metric such as an amplitude of a waveform or other related characteristic that is capable of being imparted to a photon, or alternatively another particle such as an electron.

[00215] Indeed, there is such a burgeoning variety of quantum states producing and manipulating technologies within quantum computing that it is likely that further cross-fertilization may occur with quantum networking and the ability to transmit analog quantum states information over a harness or optical network. So that within the lifetime of this patent perhaps one or other of a data carrying channel that may be isolated from its corresponding key carrying channel and vice versa may use a quantum technology that is presently unknown. Which alternates and variants are intended to be within the scope of these subject technologies.

[00216] Figure 12 is a systems flow diagram. It shows how two systems or devices may operate together to create and operate according to a randomized interface creation protocol that is provided by a process implemented as a program running on both a first system "A" 3001 and a second system "B" 3002. Provided both systems have the necessary attributes they may be considered to be compatible. But the systems do not have to be identical. The system comprised of these two systems working together may be run on any two or more compatible coupled and cooperating systems. Furthermore, the program may reside in firmware and run on a computer processor unit and operate upon flash memory within a networking card in either system, and / or it may reside in a hard drive and run on a computer processor and operate upon a random-access memory within a device such as a personal computer, or mobile phone or tablet. It might also similarly run in the apparatus of a router, and even with its processing, memory, and storage in different physical locations such as over a network.

[00217] So that it should be understood that these subject technologies are not tied to any particular configuration of hardware, so long as the available hardware is capable of cooperating to perform the task of creating the transient randomized interface protocol language; then implementing and using it to allow the two systems “A” 3001 and “B” 3002 to communicate – using the transient randomized interface protocol language they have created for encryption and decryption. Then after a random period of time (its period of transience), one or other system may generate a randomly timed reset signal or “ping” which uses the current transient randomized interface protocol language, in communications while a new transient randomized interface protocol language is created the same way as the first. This random resetting and creation of new transient randomized interface protocol languages may continue until the communication link is terminated.

[00218] It should be understood that once two systems have created a common transient randomized interface protocol language, they may resume using it at their next connection, or they may start-over and generate a new transient randomized interface protocol language. This may be done all at once, or on a rolling basis, while communications continue using the old, or the evolving transient randomized interface protocol language as new pointer-values, or pointers to characters replace the old ones. The process may more securely protect the changes to the language if this is done on a rolling basis while communications continue even as the language is evolving. It constitutes a new language when just one of its values is changed as part of such a rolling process. This aspect is developed further in the combined systems implemented in Figure 14. That may also be randomly timed 3096.

[00219] Furthermore, it should also be appreciated that using a transient randomized interface protocol language (TRIPL) according to these subject technologies does not preclude the use of any other encryption. So that there is no need for any loss of security when moving to or adding time randomized interface protocol language encryption (TRIPLE) to an already secured interface. Because the TRIPL interface can run and generate the TRIPL underneath otherwise encrypted communications if so desired; so that communications may be otherwise and TRIPL encrypted simultaneously.

[00220] For example, a TRIPLE based interface or secure messaging service may run within a webpage or browser-based extension or add-on application. So that it is running under and nested within HTTPS type encryption operated by a website, or by the browser. So that banks and other organizations may incorporate analytics adjusting ciphers (AAC) 28 and / or TRIPLE into their websites, and mobile phone-based applications. Which should confuse hackers who may crack, for example, a HTTPS or SSL protected data stream overlaid onto an AAC and / or TRIPL encrypted data stream. A TRIPL encrypted data-stream may also be used as-is without any other encryption.

[00221] TRIPLE systems may or may not be operating underneath any other encryption, and / or behind a firewall. Such additional detail is not shown. What is shown is that system "A" 3001 initiates the creation of the TRIPL with a first signal that may be called a ping 3005. The initiation signal or ping travels over any of the following Wi-Fi, Blue Tooth, Local Area Network, Wide Area Network, Internet, or another channel 3010. Which may also be conducted through a single channel along with data, or via a separate key channel 8 according to an aspect of the DICE subject technologies as explained in relation to descriptions of the earlier figures. Which channel may or may not already be secured by another encryption. For example, the two systems could be in a shared home environment, or a battle space, or even between earth and a satellite. TRIPLE may be implemented across domains, so long as the basic hardware apparatus and software are correctly installed and operating normally.

[00222] When system "B" 3002 receives the initiation signal or ping from system "A" 3001, then the system or its user may decide to accept or reject the initiation of the TRIPL creation process. Either an artificial intelligence or human intelligence may decide whether or not to accept the request to initiate creation of the TRIPL based communication (those details are not shown). Maybe this option is in a preset menu, the choices of which may be automatically set to accept or reject TRIPL creation requests. If the decision is to reject the request, then the communication is terminated 3020. Again, there is no need to see the details of the termination.

[00223] If, however, the initiation request is accepted, then this is notified by way of a feedback signal 3021 from system "B" 3002 that goes back to system "A" 3005. System "A" 3005 then decides to initiate the program 3025 to create a TRIPL for use in the combined system comprised of system "A" 3001 and system "B" 3002 working together to create, agree, and share the TRIPL. So that the system can create and run time randomizing information processing language encryption (TRIPLE). The program is set to run a repeating loop 3030 until the arrays (in the memory of system "A" 3001 and system "B" 3002) are fully populated with the values comprising the TRIPL.

[00224] The actions within the loop 3033 are as follows. System "A" 3001 creates a pointer 3045 to a random location in the seed data array 3050, then system "A" 3001 records that pointer 3045 in the TRIPL pointer array 3060. System "A" 3001 then also sends the same pointer value to system "B" 3002, and system "B" 3002 records the pointer value 3070 into its own TRIPL pointer array 3075. So now the first pointer value is stored in the first locations of both the TRIPL pointer arrays of systems "A" 3001 and "B" 3002.

[00225] System "B" 3002 now responds by randomly creating a pointer 3080 to a location in its seed data array 3085, system "B" 3002 records this pointer 3085 into its TRIPL array 3075, and system "B" 3002 sends the same pointer value also to system "A" 3001, 3090. System "A" 3001 records the pointer from system "B" 3002, 3055 into its local TRIPL pointer array 3060.

Then the system tests to see if the pointer arrays are full 3040. It only needs to test the local pointer array of system “A” 3001, 3060 to see if there are any spaces left in the array, or if a counter has reached a preset target value with each write operation. If the pointers array test returns the answer of full (or equals one), then the loop may terminate 3020 because the TRIPL pointers key will be complete for both systems “A” and “B”.

[00226] In this example, this is the first time through the loop, so the test will return a logical “no” or zero value, and this will feedback to cause the loop to run again and to generate a second pointer value that is again duplicated in the pointer arrays of systems “A” and “B”. The loop will continue to allocate pointers to seed data values or data objects until the logical test returns a “true” for the logical test of whether the pointers array is full. This can be a test on the array or may be determined by using a counter of a variable values returning a test value as being equal to the desired number of pointers required.

[00227] Persons of skill in the art may appreciate that in the creation of the TRIPL pointer array that it is filled in the sequence in which the process occurs over time. This characteristic enables TRIPL arrays to be indexed from their first value and first index location to their last. A counter may be used to cycle through the values of an array, and the counter may correlate to the index location with or without there being a parallel index array per se. From a human perspective this process would seem almost instant provided there is a reasonably good communications channel being used.

[00228] Furthermore, where a seed data value or data object is pointed to at random for inclusion, a logical test may be performed to check to ensure this seed data element has not already been used; and if used another seed data element or data object may be chosen. The size of the seed data set will affect the probabilities of two identical pointers to any individual data element of the seed data. Similarly, the complexity of the TRIPL generated will affect the size of the TRIPL. Where for example a seed data element has a value capable of conveying a meaning such as a whole word, or perhaps even a paragraph rather than a mere character, then a huge amount of information may be conveyed with meaning, using no more than a few pointers.

[00229] Once the TRIPL is created it may be used for a period of time 3095, which time periodicity may be randomized. The periodicity of these timer triggered re-set 3095 events may be varied randomly, and the minimum and maximum periods for the range of such periods may be varied according to the preferences of a user and / or system designer or manufacturer. TRIPL may be created all at once, or where a TRIPL exists already, they may be gradually replaced one pointer or data object or character at a time in the background on a rolling basis while communications may continue and adjust to the new values. As provided in the operation of an implementation of TRIPLE according to Figure 14, which provides for randomly timed pauses between loop cycles to provide a constantly morphing TRIPL 3096. Security may be further improved by the use of an AAC capable of adjusting the statistical frequency of the occurrence of members of a population (P) of characters or data patterns to make it harder to infer or guess their values. As shown in the graphs of Figure 6, 3156 of Figure 6A.

[00230] Imagine for an example application that an intelligence agent has a mobile phone with a TRIPLE application to allow secure communication by TRIPL encrypted speech. He may call his colleagues at headquarters. Speech in both directions on the call may first be converted into text, then the text run through a basic TRIPL to encrypt it into a stream of pointers. The pointers address text characters capable of reconstructing their speech in each direction. The information may be received in close to real time, at both ends via synthetic speech. Which also has the benefit of disguising the voices of agents on the call in both directions. Anyone trying to intercept this message in transit will get nothing other than a stream of pointer values, which without the seed data set and the TRIPL used should be useless to them. This is a fully time randomizing information protocol language encryption (TRIPLE) implementation.

[00231] Furthermore, in the above example an agent's actual voice may be sampled, and if the identity of the agent is known, a synthetic version of his own voice may be used to output his speech at the receiving device.

[00232] In such specialist applications, seed data sets and TRIPL may be created and installed before the two devices are used. For example, if the CIA wished to include a whole series of fixed protocols into a TRIPL as a hard-wired preinstalled set of meaningful data objects, and / or to use an AAC to boost security?

[00233] This may be done before the devices are deployed in a live operation. There may for example be emergency protocols that can be called by the TRIPL which contain comprehensive instructions, or even instruction manuals. Which may be called as part of TRIPLE communications. Thus, avoiding the need for the protocols and their detailed instructions to ever pass over potentially hostile networks. So, for example a meaningful data object may be called up by a pointer to its location. A meaningful data object within the protocol may be a text string that says: "*Abandon mission, destroy the equipment and await extraction.*" Such sensitive protocols may be stored in memory modules that are capable of self-destruct.

[00234] Similarly, US Army and Marines may use these technologies to communicate in close to real time, via speech over radio transceivers. Computational communications may also be treated similarly. But they are not covered in detail here because they may be less readily comprehensible to humans in binary or assembler codes or other machine languages. As for the languages that artificial intelligence may eventually produce using these TRIPL technologies and variants. These might be even more difficult to explain to humans in cases such as AI to AI variants. Which is why the Inventor has tried to use examples of the subject technologies set in human scenarios. It should be understood that these examples should not be interpreted as restrictive of these subject technologies and methods of operation to any one class of communications, or apparatus.

[00235] In order to try to keep the TRIPL creation process simple enough to explain one possible embodiment to illustrate its principles, we may assume a base language of this system is English, or other European languages that use the same basic character set and the numbers zero through nine in base ten and some math symbols. This approach is shown in Figure 13. But in reality, it should be clear to persons skilled in the art that any language, or combination of languages, number, or symbol base systems may be used. Indeed, the complexity of the TRIPL can be vastly increased to suit a specific task.

[00236] As explained above battlefield communications via voice, a simple speech to text conversion may be used to put the speech into a form that can be simply TRIPL encrypted, and a synthetic voice may be used to convert text back to speech in both directions. So that this simple TRIPL might suit well a battlefield radio communications system, for close to real-time secure voice communications. More complex versions may be created for handling image data and targeting data. Furthermore, in the case of artificial intelligence embodiments, the artificial intelligence at either end may be permitted more freedom to decide how they will agree their own TRIPL. Data may in all cases be reduced to binary for digital communications, and binary representations of more complex bases and data objects.

[00237] Where two systems do not begin with a shared seed data set, they may create one similarly or use a data set from a source they can both accesses. For example, an English and / or

French and / or Greek and / or the fictional "*Klingon*" language as created by the Star Trek franchise, or any other language dictionary could be read into an array of seed data and used. Using more complex data objects such as words and even whole protocols that need only a pointer value to convey their entire contents and meaning, may be extremely efficient, and extremely difficult to crack without access to the seed data.

[00238] There is nothing to stop the two systems "A" 3001 and "B" 3002 of Figure 12 or a similar more complex variant using an online source such as Wikipedia, or the collected works of William Shakespeare as their seed data set. Where for example a pointer points to an entire protocol or complex meaningful data object. Consequently, intelligence agencies might find TRIPLE useful in their communications and in their data storage and access facilities. Say for example a pointer value points to a protocol of what to do in a particular situation, and all that is needed to convey all this information is a pointer to that one single location.

[00239] Seed data sets may be based on binary, or any number base system such as hexadecimal, and image and data handling formats. The complexities of which may be proprietary in some cases. Certainly, some seed data sets may be copyright protected works. Even with potentially incomprehensible artificial intelligence created TRIPL. The essential features of a TRIPL being negotiated and agreed between two systems and expiring 3095 after a random time period. Then the next TRIPL being created using the current iteration can still be applied across applications and devices. This means that eavesdroppers will not know when the TRIPL is replaced according to the randomized timer 3095 periodicity or rolling basis and consequently eavesdroppers should always be chasing an evolving target. This may make it not worth the time and effort needed to crack any one iteration of a TRIPLE interface. Because by the time they might be able to crack one TRIPL, it may have been replaced. Probably never to be used again. So that real-time communications in particular may thus be protected using these subject technologies.

[00240] Figure 13 uses a very basic character set or database of meaningful data objects. The TRIPL system and method used could be equally configured to leverage bigger common seed data character sets or including words or other data objects or patterns capable of carrying more complex meanings and presenting smaller targets within a larger whole.

[00241] Turning now to the detail of the TRIPL of Figure 13, interface "A" 1605 of device "A" 1601 which is the interface of one machine or system or black-box module such as a network card or router or smart phone application that has been programmed to agree the values of the TRIPL, with interface "B" 1610 of device "B" 1602 which is the interface of another machine or system or phone application that has been programmed to agree these values of the TRIPL that is essentially similar. It should be noted that TRIPL interfaces can have their own bespoke CPU and enough randomly accessible memory (RAM), and firmware to store and run the TRIPLE program independently of the machines that they may serve. For example, TRIPLE capable network cards may find favor with PC and Apple Mac users, or be connected through a TRIPLE enabled router, hub, or cluster.

[00242] TRIPLE may be particularly useful for peer to peer, and local device to device connections and any combination thereof. For military battlefield operations because TRIPL operate between two or more nodes, they can vary for a data block passing over a wide area network, so that a different TRIPL may be used between different pairs of nodes. A key benefit being that in war fighting operations the time and effort required to crack any one TRIPL is not worth the effort, because no sooner has one TRIPL been used for a random period of transience – then another replaces it. So that hackers are always chasing a moving target where TRIPLE is in use. On top of which channel-hopping over wide bandwidths can be used.

[00243] Furthermore, there is no incompatibility with adding another type of encryption before or after data is passed through a TRIPL. TRIPLE may be implemented in software, or in hardware such as network cards. It is hoped that network cards will be modified to include the TRIPLE program and enough processing power and memory to use them effectively. But these processes could run on a PC or other system and still be output by regular networking cards and apparatus. This is because the technology is capable of being implemented on a simple slot in card, like a network card for PCs that requires only a modest local CPU, an amount of on-board RAM and firmware to store the program code to run it.

[00244] TRIPLE may run on existing hardware with existing interfaces and network cards etc., by utilizing just a small amount of the local device CPU's processing capabilities, RAM, and storage. So that TRIPLE interfaces can be run on many of the smart devices as they already are by deploying them as Apps and / or as web- browser extensions and / or running them from within webpages for extra security.

[00245] The new level of security that can be added by the constantly changing and evolving TRIPLE is also likely to be very useful for military applications for secure networks. That an adversary will not know how to crack because monitoring the traffic does not help when the TRIPL is constantly changing.

[00246] When data moves over longer distances it may pass through multiple TRIPL so that trying to study it along the path should cause confusion as the adversary eavesdropper will be looking at different looking data intercepts when the same data passes through multiple TRIPL. So that even if they can see the same data at multiple locations, they may not be able to know or not realize they may be looking at the same data but differently encrypted.

[00247] Even if one TRIPL within TRIPLE communications is cracked it will probably get replaced before much damage can be done. This constantly changing feature of TRIPLE should be capable of allowing war fighters enough real time protection for relatively secure battlefield communications. That cannot be cracked quickly enough by an adversary to compromise time sensitive tactical information.

[00248] Interface "A" 1605 of device "A" 1601 agrees the values of the TRIPL with a compatible Interface "B" 1610 of device "B", within a feedback process. The creation of the TRIPL is triggered at random time intervals, by a time module present in both interfaces 1642. The interfaces by a feedback process 1615 agree the meaning of the characters, words or other meaningful data objects that comprise common seed data. For example, after the reset "ping" 1640, Interface "A" and "B" may take turns to randomly assign characters to numbers, or they could use other ways of selecting agreed values all at once, or on a rolling basis or all at once after the first TRIPL has been created.

[00249] In this example an array is created in both interface "A" and interface "B", where number values or pointers 1625 correlate to one of the characters in the array of meaningful data objects from the common seed data 1620. To create the agreed values 1630 for the TRIPL. In the example it can be seen that the value or pointer 1 correlates to the character "h", and the value or pointer 2 correlates to the symbol "&", and value or pointer 3 correlates to the symbol "]", value or pointer 4 correlates to the character "s,", value or pointer 5 correlates to the character "P", value or pointer 6 correlates to the symbol "$", value or pointer 7 correlates to the symbol "+", and value or pointer 8 correlates to the symbol "?", with the remaining symbols and characters yet to be assigned as the two interfaces create their agreed character set by assigning numbers to them and vice versa as interfaces "A" and "B" cooperate to match number values or pointers to characters.

[00250] Once they have finished assigning values to meaningful data objects, then they have a common agreed interface protocol in which they may communicate in any language or mix of words from any languages that can use that character set. So that all characters, symbols, and bigger more meaningful data objects are represented by their agreed values and communications are achieved by exchanging streams of these values or pointers. Until the randomly timed 1642 reset ping is triggered again 1640, to initiate replacement of one or more or all of the TRIPL values or pointers.

[00251] Whereupon the values of meaningful data objects are randomized and agreed again to the next iteration of a constantly evolving TRIPL. The meaningful data, comprising seed data may be any data object. For example, the entire contents of a dictionary can be given agreed values, multiple language dictionaries can be used, and words of the same meaning can be randomized between similes and equivalents from different languages. So that agreed word-1 may be French, agreed word-2 may be German, agreed word-3 may be Navajo, agreed word-4 may be Spanish, and the TRIPL will create this hybrid transient language for use over a limited time period after which another language is created and so on. So that it is too difficult and / or not worth the effort to try to crack any one TRIPL. Because no sooner has it been cracked than it has been replaced never to be used again.

[00252] The maximum seed data that is potentially usable may be as large as the sum of all knowledge that is in electronic form. So that the TRIPL might agree to use a novel as seed data, wherein agreed words are taken by page number, line number and from left to right for example. So that both TRIPL agree to use one or more eBooks as seed data. Furthermore, number bases may be randomized for math functions. So that random number bases may be used for calculations in a similar way. The possibilities are infinite. Star charts could even be used as seed data. This example has been kept relatively simple. Once all the TRIPL agreed values are all set, then the agreed values or pointers to the agreed values can be read into an array, or array of records so that meanings can be attributed to all the agreed values while the language is used. More complex data such as images and sounds, such as spoken text can be used similarly. So that two computers with internet access may even select their seed data randomly from the Internet so long as they can agree the meanings of the agreed values they use, the possibilities may be infinite for the TRIPL that can be created used and then deleted without anyone ever even knowing what the TRIPL was. These subject technologies may include seed data objects, and languages as may be used and created in part or entirely by artificial intelligence to provide TRIPL, for use within TRIPLE that may be utterly incomprehensible to humans.

[00253] Figure 13A is the TRIPLE interface between two nodes, where the process internally is not knowable on the outside and there is no need to know the details of the TRIPLE on the inside. So that from the user perspective the TRIPLE interface is a secure black box that may operate over local area networks. The complexity of the TRIPL used internally can be scaled to match security needs in specific applications.

[00254] Figure 14 is the flow diagram of a process for creating an agreed TRIPL which operates on two cooperating systems “A” 3001 and “B” 3002; and is capable to continuously evolve over time. An isolated channel for communications used in creating the TRIPL that is operated according to DICE may or may not be used.

[00255] This is achieved by operating its TRIPL creation process as a continuous loop which continuously refreshes the pointers arrays 3034. TRIPL creation operations may be used after such a system as that of Figure 12, or Figure 13 has already populated the pointers arrays, or it may be used from beginning to end.

[00256] Much the same functionality may also be achieved by adding a randomly timed pause between the looping processes of those alternatives, that may be triggered via a counter or other logical test once the pointers arrays have been initially populated. The key feature being to make whichever implementation to cause the TRIPL to be renewed on a rolling basis, by renewing one character or data pattern. Then pausing the loop for a time period that may be set to operate to pause the loop for random periods of transience between a maximum and a minimum parameter.

[00257] That periodicity and its randomization features may also be parameters that may be preset or set by users in some alternative embodiments. A non-randomized periodicity could be used. However, adding randomization to the periodicity may help to make decryption more difficult for codebreakers. Detail of how those features of timing parameters may be set may vary according to preferences, details of which are not shown.

[00258] Because codebreakers seek to identify repeating characters or patterns, for use as a crib or clue and a way in. By which to begin to crack ciphers and encryption, it makes sense to use TRIPLE because of the transience of each TRIPL. This finite opportunity to sample any TRIPL before it is replaced limits the amount of data that can be used to break the encryption based on that specific TRIPL to its period of transience. The length of which is a feature that may be randomized to helpful effects. Furthermore, the period of use of any one TRIPL may be made very difficult to parse from the period of transience of another TRIPL. Because a replacement TRIPL may be created while using the predecessor. That is the basic form of operation of TRIPLE.

[00259] Furthermore, by adding replacement of the TRIPL's individual characters or data patterns one at a time followed by randomly pausing the looping process for replacement on a rolling basis, during another randomly timed period 3096. Then every time one TRIPL character or data pattern is replaced, the TRIPL morphs into a similar but non-identical TRIPL. Causing yet more headaches for codebreakers who will be unable to easily see where one TRIPL is replaced by another. Because the process may be thus blurred over many similar but randomly different TRIPL extending over time. During the period of the pause 3096, the current iteration of the TRIPL may be used, and so on.

[00260] Furthermore, a character or data pattern may be re-selected over time but attributed a different value than in its previous period of transient use. To further confound codebreakers. Because they may be looking at data in which the same character or data pattern comes and goes and has different values during each of its periods of transience. It is highly likely that codebreakers may be driven to experiment with some extreme mathematical techniques to try to find a way to crack such an evolving TIPLE message, or document, or data stream etc.

[00261] Assuming this TRIPLE process is taking over from another system that created a TRIPL, or perhaps it has been pre-installed, or shared and is one that is already being used. Then a computer processor of that system may run this module having bypassed earlier steps to begin execution at step 3034 to continuously loop to refresh pointers arrays.

[00262] Alternatively, a TRIPLE process according to Figure 14 may be used from the outset and the pause timer step 3096 bypassed 3061 during the creation of the first TRIPL. The process for populating the TRIPL pointer arrays "A" 3060, and "B" 3075 may be identical to that provided in Figure 12 and is not further explained here for that reason. This implementation differs in the incorporation of the randomly timed pauses between loop cycles that may provide a constantly morphing TRIPL 3096. Which is suitable for use in operations once the first TRIPL has been created in the pointer arrays. The feedback 3033 may be a logical test condition which may be used to determine whether or not to bypass 3061 the pause timer 3096 in the loop process 3034.

[00263] If either system is caused to close down or end the communication session a signal and / or lack of signal 3031 may trigger the termination of or break the loop 3032. So that there is provided a neat process which does not consume power or system resources when not needed.

[00264] A new communications session and the creation of a new TRIPL or use of an existing stored TRIPL from a previous session, may be begun if either system takes the first step to become System “A” 3001 by initiating a new dialogue with a ping, over a communication channel 3010 to a second system “B” 3002. Which may accept or reject the request 3015, by providing a feedback signal 3021. Then if the second system “B” accepted the program module may execute 3025 and trigger creation of a TRIPL, by calling the TRIPL creation looping process to run the continuous loop which continuously refreshes pointers arrays 3034.

[00265] If this is a first boot TRIPL creation, then the randomly timed pauses between loop cycles 3096 may be bypassed 3061 or set to a zero-time period between loops for the first TRIPL creation process. Then begin to run as normal thereafter once the TRIPL arrays are fully populated.

[00266] Figure 15 is an expanded view inside a black box TRIPLE such as system the one provided by Figure 13A. It shows Interface “A” and “B” 1690 as being identical. But they need not be identical in their details so long as they are compatible. Here we see that interface “A” belongs to and is part of and is running on phone device “A” 1647, and that interface “B” belongs to and is part of and is running on the Device “B” 1649. The program or application can be provided in software and / or embedded and / or installed into firmware 1650. The program runs on the local CPU 1655 and uses the local working memory or RAM 1660, it may be stored and bootable from local firmware or flash memory 1665, and capable of operating on the various forms of seed data characters, and / or dictionaries 1675, and / or known languages in text and / or speech 1670. Which seed data sets may be capable of updates and / or expandable in some systems to a maximum equating to the available on-line sum of human knowledge 1680; so long as the data is in a networked electronic form connected to the TRIPL interface.

[00267] The newly created TRIPL data may be stored during its use into either the flash memory 1665 or held in working memory during operation. So that once activated the programs 1650, in both interface “A” and “B” using those resources, use a protocol and feedback process 1695 to create a succession of new TRIPL for use during random length periods of time in TRIPLE communications. That facilitates secure TRIPLE protected two-way communications between interface “A” and interface “B” 1699. They use a language only they know that will exist only fleetingly during use, after which each will be automatically replaced by a new or evolving TRIPL iteration.

[00268] Users, however, do not need to know any of that, they just need TRIPLE capable network cards, or devices running a TRIPLE App, or other compatible variants to gain end to end encryption with no keys per se, just a succession of secure disposable languages that are transient in nature and mostly not worth the effort to hack.

[00269] The principles of a basic TRIPL may be augmented by using AI. These may be in binary, or any number base system, and they may ascribe meanings to data objects that may or may not make sense to humans. But the TRIPL they may create as between themselves may stay between themselves. It may never need to be or be communicated outside their closed loop, no record of TRIPL needs to be kept. The larger the data sets and seed data they are able to use, the more combinations of possible values and languages they will be able to create.

[00270] Figure 16 shows the creation of a TRIPL, based on randomized pointers 3110 to an array populated with seed data in the form of alphanumeric characters 3115, and the index location values in the array of pointers 3105.

[00271] This randomization of pointers 3110 and the values of the array 3115 may have been preset, or it may have been created in a TRIPL negotiation as explained in relation to previous figures, such as Figure 14. It may also be left over from a previous communications session between two or more devices or nodes; for use as and when they reconnect. Devices may store multiple previously negotiated or installed TRIPL from previous interactions with other devices. Their historic tally data records may also be used to inform both systems if they recognize each other. To see if they still have a TRIPL in common. If there is no TRIPL in common existing between two devices, they may create this TRIPL as a new TRIPL.

[00272] Use of the addressing mechanism of pointers means that these relatively meaningless values of the pointers themselves may be used as the basis of information exchange communications using TRIPLE. This is good because they will most likely be totally alien to and unrecognized per se, by many cryptographers, hackers and crackers who may intercept them.

[00273] Furthermore, as previously explained TRIPLE may be used in addition to existing encryption in the art, without conflict. This may permit an additional layer of deception because cryptographers may crack the other encryption only to be left with a stream of meaningless pointer values. Which it is hoped will leave them baffled and may cause them to believe that their decryption efforts must have failed. So that while TRIPLE should not be confused with the encryption available in the art. That may be boosted in efficacy by use in combination with TRIPLE. Furthermore, such double encryption with TRIPL plus HTTPS and / or SSL etc., is a new combination that may have more value than the sum of its parts.

[00274] Using an array of pointers 3110 with an index or indexable capability 3105 has other advantages in that it may be more secure than using a TRIPL based on using the randomized characters represented in this example. In the following Figures 17, 18 and 18A the distinction as between using the pointers to the values or using the values is not shown. It should be understood that though these details as provided in the system of arrays and pointers of Figure 16, are not shown they may be present, or the values may be being used directly.

[00275] Figure 17 shows some extracts from the same values for the array index 3125 locations “1” through “9” as reproduced in simple form from those of Figure 16. Here again the first index location translates or points to the alphanumeric character or value “3”, the second index location translates or points to the alphanumeric character or value “j”, the third index location translates or points to the alphanumeric character or value “u”, the fourth index location translates or points to the alphanumeric character or value “<”, the fifth index location translates or points to the alphanumeric character or value “y”, the sixth index location translates or points to the alphanumeric character or value “g”, the seventh index location translates or points to the alphanumeric character or value “e”, the eighth index location translates or points to the alphanumeric character or value “k”, the ninth index location translates or points to the alphanumeric character or value “a”.

[00276] The figure breaks off the sample of alphanumeric characters or values after the ninth index point and resumes at the hundredth index location. Where the values contained in the array have begun to include more than single alphanumeric characters, and the array contains strings of alphanumeric characters or values 3135. The hundredth index location translates or points to the alphanumeric character string or value “and”, the hundred-and-first index location translates or points to the alphanumeric character string or value “car”, the hundred-and-second index location translates or points to the alphanumeric character string or value “computer” and so onward.

[00277] Again, after more of the TRIPL strings or values the figure breaks off the sample strings or values; and resumes at the two-hundredth index location. Where the values contained in the array have begun to include more than one-word strings of alphanumeric characters or values

– and are comprised of longer strings of characters or values that represent multiple words, and or database records comprised of complex data that may even be comprised of data of disparate types 3140.

[00278] The two-hundredth index location translates or points to the alphanumeric character string or value "*Initiate emergency procedure. Remove Hard Disc Drives and take them to dead drop Apache.*" The two-hundred-and-first index location translates or points to the alphanumeric character string or value that comprises a link to another data object which is *"SATELLITE IMAGE FILES*". This may be a composite data object that may include strings of descriptive alphanumeric text, hyperlinks, and image data files, perhaps within a database record type of structure. The two-hundred-and-second index location translates or points to a composite literary work, or booklet, or database called "*USMC / Army Field Manual for RPG*". The two-hundred-and-third index location translates or points to an instruction to switch the case of an alphanumeric character "*Switch case to capitals code 0110*". The two-hundred-and-fourth index location translates or points to the alphanumeric character string or value "*Rocket engine signature suspected ICBM. Emergency scramble intercept.*" and so the sample goes on, but further details are not shown in this illustrative sample.

[00279] Figure 18 shows in detail like a sample of program code, the defined process comprised of a run-time example of one possible TRIPL creation processes 3150. Which details the run-time decisions and actions of a pair of devices cooperating to create the TRIPL of Figure 16, and which may also be equivalent to or if implemented as an array of pointers for the TRIPLE application, or its value-based variant of Figure 17, and the further data object examples shown 3135, and 3140 of the examples. Please see the defined process 3150 for its specific run-time actions in detail, as these are very illustrative of the processes of Figures 12, 13, 13A, and 14. It should be appreciated that there are many possible variants that may be used here that are intended to be within the scope of these subject technologies. Probably the most weird and difficult to understand are the ones that may be created by artificial intelligences operating these systems and methods. Some of which while within the Inventor's contemplation are impractical for use as illustrative tools. Artificial intelligence may come up with surprising variants, without deviating from these subject technologies.

[00280] Figure 19 provides a local system or device 111 which may be a PC / or phone / or pad / or missile / or unmanned aerial vehicle / or other system. Having a central processor unit or controller 100; that runs the local randomizing and / or reassembling application (program) 105. Which presents options 110, and the user may select files and options 115, via the user interface on their device 111. These selections are fed back 120 to the local randomizing and / or reassembling application. So that the application can work with selected files in the local random data storage locations 142.

[00281] Which may have an auto-eject mechanism (of a data card or other storage apparatus) or auto-destruct capability131. That may be helpful in defense systems applications, where for example a drone or unmanned aerial vehicle (UAV) may be at risk of capture and interrogation. Furthermore, there may be other storage locations that may be, configured to be separate and used for the storage of a pointers based key array 144. Which may be isolated 14 from the data storage locations.

[00282] So that the DSURF encryption of the data may be impossible to reverse if the isolated key storage 144 is securely deleted (according to a process such as that shown Figure 35). Additionally, the local key storage 144 may have an auto-eject (of data card storage apparatus) or auto-destruct capability131. That may be helpful in defense systems applications, where for example a drone or unmanned autonomous system (UAS) may be at risk of capture and interrogation. These features may allow a user to encrypt and safely store their data locally according to these subject technologies.

[00283] Furthermore, if the isolation 14 of data locations and channels “A” 152 from keys “B” 153 is operated in combination with dual isolated channels encryption (DICE) a defense system on a sensitive mission inside hostile territory may securely provide its DSURF encrypted data securely to the American defense Cloud, and / or to an operations command-and-control node. By using communications channel 1 and channel 2, that are configured for use with DICE . Configured for compatibly operating DSURF and DICE. Which may allow the randomized data to be securely uploaded and distributed similarly into random data storage locations 142 in the Cloud, over a channel 152 that is separate and isolated from another channel 153 which is used for key storage of the pointers array key storage locations 144 which itself is isolated 14 from the random data storage locations.

[00284] The local device central processor unit controller 100, also may have the capability to operate across a connected local area network (LAN), wide area network (WAN) or Cloud 135. So that it may also act upon selected files 115, in the further alternative remote storage locations 142, and the arrays of pointers keys may be either kept locally or in the alternative cloud key storage locations 144. So, the data and the keys may be isolated from each other by one or other of these possible combinations by ensuring they are not located in the same storage system or drive or cloud storage location.

[00285] The communications channel(s) channel 1 and or channel 2 between a local device and the server side network 104 may use multiple separate channels. Or they could use separate “threads” and / or different “ports” of the local device. On up to the cloud 135, where they may also be isolated from each other 14, with data being stored and accessed in random storage locations 142, via isolated network channel “X”, 152, and arrays of pointers-based keys stored in a separate and isolated key store 144 (which may be in a separate cloud infrastructure, and / or even owned and operated by a cloud-based separate key storage provider. That may be accessed via an isolated and separate channel “Y”, 153. According to similar and compatible aspects of both the DICE and DSURF technologies.

[00286] The alternative remote storage locations 142 may be among many servers and connected to the controlling server by a communications channel “X” 152, being under the management of a remote randomizing and / or reassembling helper application 155, running on a remote server 151 or multi-server management system having a central processor unit controller(s) 150. So that either locally or at the server level files may be broken apart into smaller data blocks, and those may be randomly allocated into the various data storage locations 142, which are isolated 143 and distinct from the pointers arrays or key storage locations 144, and which isolation may include a separate communications channel “Y” 153. This is so that if hackers gain access to the cloud data storage locations 142, the keys needed to find, download, and resemble the necessary data blocks to recreate the original files are not co-located with the data blocks. Leaving hackers unable to reassemble any useful data from the storage locations 142. The pointers array-based keys may be kept by a separate entity in highly secured Cloud based key storage locations 144, and accessed via a separate and isolated channel “Y”, 153. This use of separate isolated (unbridged) communications channels “X” 152 and “Y” 153, and separation and isolation of the data storage locations 142 from the arrays of pointers or keys 144, needed to find and reassemble the data blocks back into usable files is addressed in more detail in Figures 41 and 42 below.

[00287] The file server 151 may have multiple networking cards and / or connections ports such as the RJ45 standard, and any future or other variants which connect the server CPU 151 to the storage locations for data 142 and pointers array keys 144. In very simple small-scale implementations, the data storage may be integral to the server provided it can be isolated from the key storage locations. In most large-scale applications as may be operated by larger cloud storage and processing providers, it is more likely that the data storage will be accessed via an RJ45 connection type routed networking architecture within the cloud infrastructure.

[00288] The server 151 may have two such RJ45 connections or similar connections to provide two isolated channels capable to support two unbridged (unconnected) and hence isolated 143 communication channels “X” 152 which is connected to the random data storage location infrastructure 142, and which channel is isolated 143 from a second channel “Y” 153 that is coupled to the pointers array key storage locations infrastructure; which may be within an entirely separate specialist key storage cloud infrastructure and may be owned and operated separately and independently from the random data storage locations 142 cloud infrastructure.

[00289] This separation and isolation of randomized data 142 and data communications channels “X”, from pointers array key storage locations 144 and the isolation of these networks from each other using data channel “Y” 153, is to ensure that hackers who may breach the data storage infrastructure, should not be able to find or access the keys needed to reassemble the files, or user account data stored there.

[00290] Using cloud-based storage for randomized data, and pointers-based array keys should not be appreciably slower than existing cloud data storage and processing systems, nor require more processing power or bandwidth. Indeed, it may even be less demanding and hence faster in use than some math-heavy forms of file encryption. In relation to which server-side operations may be performed by experts and may be subject to the presentation of options to the server administrator 160. Noting that the server administrator 165 may be human or an artificial intelligence. The server administrator may thus provide feedback 170 to the remote randomizing and / or reassembling or helper application. This interacts with the remote storage locations to serve the needs of the user system. Which may communicate back and forth to support the local operations being performed on the user's files. So that a user with good connectivity may have a seamless experience that varies very little as between the storage locations used regardless of whether they are local, or cloud based.

[00291] This model of operation spans simple local hard drive operations all the way up to running multiple separate physical channels or threads (used as separate channels) to multiple cloud-based data storage locations 142, and specialist cloud based key storage 144, that is isolated from the data and accessed server-side via a different communications channel. All of which possibilities are covered within this one flexible example. This embodiment is capable of being so flexible due to the growing levels of inter-connectivity and compatibility that is becoming ever more available to users across devices and networks. More detail including the steps not shown in this figure and its server side parts and operations 104 is shown in subsequent figures. In particular in Figures 41 and 42.

[00292] However, many users of modern smart devices seem not to care sufficiently about securing their devices and files to bother with encrypting them locally. So that perhaps this system may be preferred by people with secrets to keep such as defense contractors, and users in military or intelligence applications. Where the device itself might be at risk of falling into the hands of an adversary or being compromised by hackers.

[00293] Whereas an alternative approach may provide similarly improved data security that is handled mostly or entirely within the cloud. Behind a black-box interface that users do not need to know even exists or care about. So that a similar consumer user system may leverage the benefits of these technologies. This may be made indistinguishable from existing systems from the outside of their account interface, with AWS, or Google Drive, or Drop Box, or Microsoft One Drive etc. This server-side processing option, and perhaps a user invisible variant of this, might offer an easier way to roll out the benefits of these technologies for most consumers. That would best fit the current trend towards keeping most user data in the Cloud and with the rise of generative artificial intelligence this may further expand to include also doing more processing in the Cloud 135.

[00294] Local files such as email files are often compromised in local storage by hackers, via Trojan Horse attacks. So, there is merit in keeping all that sensitive user data exclusively in cloud storage, according to these subject technologies. In the hands of experts such as Google so users can relax and let experts keep their data safe using these subject technologies. This lazy user paradigm, off-loading this task to experts, and expert systems including AI that may operate these subject technologies in the Cloud appears to be the option most likely to be preferred by most users! Ergo, also the most commercially viable and valuable way to implement these subject technologies.

[00295] Many users of services who may adopt these technologies such as banks, social media, and medical care providers may use them to prevent mass data theft events. Without their users being aware of the technology upgrade in data security. Which is why the next figure is addressed more particularly to DSURF server-side applications, and / or the lazy user who does not want to operate his or her own data security systems locally. But who prefers to entrust that role to the experts who run and maintain the Cloud.

[00296] Figure 19A provides such an improved cloud based invisible embodiment of these subject technologies. That is tailored to mass market users, including unknowing users at the local device level. Which may use a website or web-application 106 via communications and data channels (channel 1 and / or channel 2) to access and use a version of these subject technologies that may be running in the cloud or another other remote network such as an intranet, and which may operate DSURF within the Cloud infrastructure 135 within the server side network 104 like a black-box 1497 (further details of which Cloud-based black box operations are provided for uploads in Figure 41A, and for downloads in Figure 42A).

[00297] Although this embodiment may look very similar to the embodiment of Figure 19 above this is embodiment is the most likely to suit American consumer users of mobile phones, tablets, and computers. It may even use these technologies, running inside a cloud / server-side black box without the knowledge of end users . For applications such as online banking and cloud drives for example.

[00298] A device such as a personal computer or smart phone or pad or other computing device 113, may be a running remote web page and / or a web-application (App) 106. This may be cooperating with a cloud drive, or healthcare website or an online banking website etc., within a Coud infrastructure 135 through which users are presented options 112, and may select files and options to work on their data or services via a user interface 116.

[00299] Those options and selections may be obtained by the App 106, by way of feedback 121. That App may then operate to control the device controller CPU 107 according to the programming of the web-browser or application 106. Which may cause the controller CPU 107 to operate the communication and data channel(s) (channel 1 and / or channel 2) to operate upon files, that may be stored locally in the alternates of fixed hard disc drive, or solid-state drive 125, or removable media such as SD card, or USB drive, or other media 130.

[00300] Such files as may be sent to the sever side network or cloud 104 may be received over a network and into a cloud infrastructure 135 or management system, that may operate as a server-side black box (1497 of Figure 41A and Figure 42A). So that DSURF processing may not burden the local system and may instead take place inside the cloud infrastructure. Files already in the cloud infrastructure 135 may be accessed and worked upon, and services accessed and used also via the website and / or application 106 similarly.

[00301] Data may pass back and forth between the local device 113 and the server-side network 104 of the cloud infrastructure 135. The precise details of uploads, downloads and services or computations that may be provided may vary as between various possible user applications. Which may include the use of generative artificial intelligence and other processing that may take place within the Cloud infrastructure 135. Users may access many websites and services via websites or applications that may be operated by different providers of a wide range of possible services as may be provided over a network such as a Cloud infrastructure, and / or via the Internet including all the sub-networks such as cellular phone networks, optical fiber networks, and satellite networks that may provide users with access to the Internet and the World Wide Web (WWW). All of which possible combinations would be unnecessarily onerous to list or address in detail here.

[00302] Though not shown here the secure deletion of locally held files (addressed in relation to Figure 35 and related materials below) may be an option provided by the web page or App 106. So that only the secure version of those files will exist thereafter within the cloud infrastructure 104 if so desired, for maximum file security.

[00303] Users may alternatively also wish to keep an offline backup of files in removable media 130, or perhaps in protected local storage 125. Which may have isolation between them 14, so that if it were desired to save DSURF encrypted files locally. They could be downloaded with the data stored into one of those locations 125 or 130, and so that the keys may be kept isolated 14 away from the data by being stored in the other of those locations.

[00304] This option is probably something that only security minded people and professionals would need to do. Consequently, for expert users who may be IT security professionals and government employees, including military and intelligence users of the embodiment shown in Figure 19B may be preferred. Which varies to include the isolation 14 of the fixed storage 125, as operated on coupled channel "A" 152 and isolation of that channel 14 from the channel "B" 153 and from its coupled removable storage 130 .

[00305] Here server side 104 secure cloud-based 135 storage, processing, and services are used. The files to be stored may be received via communications channel(s) (channel 1 and / or channel 2) that may or may not be used and configured also according to the DICE subject technologies. Which may include variants and differ considerably in their implementations. They may be configured as at a minimum efficacy level as different "threads", using ports on one physical channel, and for greater efficacy as separate physical channels for data and for encryption details and keys, and for example one may be via a mobile phone and the other a home Wi-Fi system. Examples of some of the possible permutations are provided in Figure 1, Figure 1A, and Figure 2, Figure 3, Figure 4A, 4B, and 4C when operated to provide the DICE, which may be operated in combination with DSURF an example of which combination is provided in Figure 2 and the descriptive materials above relating to it.

[00306] The communications channel(s) may connect to Cloud infrastructure 135 having an internal server-side network 104. That is under the management of a server 151, wherein a suitably programmed computer processing unit 150 controls a randomizing program and / or reassembling program 156 application. Which is configured to operate to break files apart into blocks of data, that may be similarly or randomly sized, and may be stored into random storage locations 142; which random storage locations of the data block may be recorded into an array of pointers to provide a key within separate key storage locations 144 and that may be separate and isolated 14 from the data storage locations 142. These keys may later be used by the reassembling program 156 to retrieve the data blocks and then to reassemble them back into copies of the original files.

[00307] A server administration user interface may be used to set parameters for available options as may be presented to their expert administrator users 160, in accord with human or artificial intelligence user input to set options 165, which input is provided as feedback 170 to the suitably programmed controlling computer processor unit 150 of file server 151.

[00308] The file server 151 may have multiple networking cards and / or connections ports such as the RJ45 standard, and any future or other variants which may connect the sever CPU 151 to the storage locations for data 142 and pointers array keys 144. In very simple small-scale implementations, the data storage may be integral to the server provided it can be isolated from the key storage locations. In most large-scale applications as may be operated by larger cloud storage and processing providers, it is more likely that the data storage will be accessed via an RJ45 standard / or equivalent optical channel connector or other possible variants and routed through a networking architecture within the cloud infrastructure 135.

[00309] The server 151 may have two or more networking cards having their own RJ45 connections or similar connections that are unbridged (not coupled to each other per se) to provide parallel isolated 14 channels "X" 152 which is connected to the random data storage location infrastructure 142, and which channel is isolated 14 from a second channel "Y" 153 that is coupled to the pointers array key storage locations infrastructure. Which may be housed within an entirely separate specialist key storage cloud infrastructure and may in some implementations even be owned and operated separately and independently from the random data storage locations 142.

[00310] For example, users may have a key storage account with a specialist service provider infrastructure, but their data storage may be provided by bulk data storage service providers such as AWS, Google Drive, Drop Box or Microsoft One Drive. Persons skilled in the art will appreciate there is no reason why a cloud-based data storage provider may not also operate their own isolated cloud based key store, and that their key store might be compatible and capable of use with data storage provided by a competing service. Indeed, such arrangements may help to diversify the possible key storage locations such that hackers are rendered even more unable to guess or predict where the key for any data block may be stored.

[00311] This separation and isolation of randomized data 142 and data communications channels “X”, from pointers array key storage locations 144 and the isolation of these networks from each other using the separate unbridged bespoke data channel “Y” 153, is to ensure that hackers who may breach the data storage infrastructure, should not be able to find or access the keys needed to reassemble the files, or user account data stored there. More complex configurations of these subject technologies are possible. These are too numerous to list but are intended to be included variants that are within the scope of these subject technologies.

[00312] More detail including the steps not shown in the embodiment of Figure 19A and its server-side parts and operations are shown in subsequent figures. Including in particular Figure 41A, and Figure 42A.

[00313] Figure 19B provides a variant embodiment tailored for security minded people and professionals. The main difference from the previous figure is that it may have a local randomizing and reassembling capability, that may be provided by web-page applications or browser extension, and / or by a locally installed application 106. Which may operate on local data in a similar way to the server-side application 104. But do so locally rather than rather than exclusively performing those operations within a Coud infrastructure 135. Wherein the fixed storage 125 is operated on one channel “A” 152 and may be isolated 14 from the removable storage 130 which is operated on channel “B” 153. So that both storage locations for encrypted data and keys may be isolated from each other. Furthermore, where the key is kept on removable storage it may be removed when the local system is not in use. So that no other user can access that user’s encrypted local data, and no hacker can decrypt them even if they have managed to mount an intrusion into the system. Which may be useful in organizations that use hot desk swapping between employees. Who may have compartmented duties that do not overlap. In which case, it is recommended to keep backups of both keys and data in secure and separate locations from the encrypted data to mitigate the risk of data decryption and theft. Secure keys may be held on a USB drive or SD card and may be dismounted and stored within a physical safe for example. Aside from which minor differences in the local software 106 this embodiment is otherwise the same in terms of apparatus and operationally as that provided in the previous figure.

[00314] Figure 20 is a block diagram showing the main blocks that may form the core of a maximum-security randomized data handling and communications system 1700. Utilizing randomized storage of various possible types and levels of storage of DSURF 1710; randomized interfaces 1720; randomized routing and / or isolation of data and keys in transit 1730; and randomized priority-based timing 1740.

[00315] Figure 21 is prior-art. Showing transparent processes predictably use resources 1750; and this is so, and less secure because data, storage, RAM, CPU, Transistors, Components and Devices; are easier or more predictable to find and therefore easier to hack 1760. This may not be a great surprise to those skilled in the art, but it should concern us all due to the massive proliferation of networked smart devices joining the “Internet of Things” (IOT). Which criminals and intelligence agencies worldwide may be looking to exploit new hacking opportunities. The majority of which IOT is likely to be operating autonomously for an increasing amount of the time. Furthermore, AI agents will be taking decisions, that impact networks, and that travel over networks. So that the Internet of the future is going to need increasingly smart security to defend all the IOT devices and robots, as well as human users. The Internet is probably destined to become, and arguably already is an ecosystem in which artificial intelligence increasingly operates.

[00316] Virus makers will probably begin to create something akin to AI bacteria and viri. Smart hackers already can guess the configurations of hardware and predict the challenges they face when hacking known systems. As their tools get smarter this problem may get worse. Which is why there is a need to increase the difficulty levels of predicting the challenges hackers face. Rather like the concept of stealthy aircraft being difficult to target. So too stealthy data, even stealthy big data may be made into a much harder target to find and / or to attack.

[00317] Figure 22 is a block diagram showing aspects of secure processing using randomized resources 1770. Where data may be passed around using a black box such as TRIPLE between nodes 1775 in a system. May provide increased security for the data passing through.

[00318] Furthermore, the use of DSURF randomized data, and / or randomized storage of data 1780 can add further security. Where signals intelligence and high-end data is being protected randomization of the use of RAM and devices and / or their communication ports can make them harder to find and hack or spy on 1790; this can be taken further with multi-core processors; or group / or graphics processors (GPU) and / or neural processor units (NPU) in the context of creating neurons from RAM and GPU.

[00319] Signals Intelligence eavesdropping technologies and methods can be used to capture data leakage from electronically noisy systems. This noise has been studied and a whole signals-intelligence analytical technology has been created to enable it to be reverse engineered back into the data that created it. Thereby allowing sophisticated signals-intelligence (Sig-Int) to duplicate the data being processed by an IT system such as a printer or computer, or network adapter card. Signals Intelligence is also a problem for US defense and intelligence agencies.

[00320] Even with this Signals Intelligence and analytical spy tech including TEMPEST it may now be possible using these subject technologies to create some parts of computers that are capable of operating and to use resources sufficiently randomly that they cannot so easily be spied upon using signals intelligence and analysis of data leakage.

[00321] Randomized processing, and randomized data handling when combined, may be used to provide more secure processing, and communications capabilities in the modern battlespace.

[00322] Figure 23 goes into more detail than Figure 20, Figure 21, and Figure 22. These three figures are intended to be fully compatible with those that follow them. Figure 23 is zoomed into the details of possible implementations of the subject technologies, and their relative merits in securing data. The options to the left being relatively less secure than the options to the right of the figure, as indicated by the left to right security level slider gradient 199.

[00323] First the files for the program and CPU to operate on are selected 172, then either the alternate process to randomize data blocks into storage as contiguous files 174 or randomize data blocks to disparate non-contiguous storage locations 176, or randomize data block sizes, then randomize files, and storage locations 178.

[00324] Whichever alternate randomization options are chosen; the next operation the system performs is to write data blocks and an array of pointers to their locations in storage 180 is executed. This can be according to the alternatives to store both files and pointers array or key in the same drive or media location 185. This is the least secure option due to the risk of hackers being able to access both the key and its associated data. Then use the former to reconstruct the latter just by hacking the local drive or storage media.

[00325] A more secure alternative is to store only the file(s) on a local drive and store the pointers key separately either in a hidden location, which can be a hidden partition, or even a removable storage device such as USB or DVD etc. 190. Probably the most secure, and convenient alternative is to store files randomized across multiple cloud locations and store the pointers key separately into specialized cloud key store 195. This may be differently and independently operated than the cloud storage. For example, with the cloud storage may be provided by AWS and the key storage may be provided by Verisign or Microsoft. This can be both secure and seamless because multiple data threads can be run over modern networks without any perceptible loss in performance from the user perspective. DICE is also increasingly possible due to increasing device connectivity.

[00326] Looking at this diagram from left to right, the further to the right of the figure the more safe, secure, and seamless and convenient the user experience may be 199. The options are also compatible with streaming data for communications and media. These aspects of the subject technologies are dealt with in more detail below.

[00327] Figure 24 shows in more detail the options to be offered by a system running a program according to the subject technologies on its central processor or group processor units, and the operations the system performs on the data. The system may pre-encrypt data 200. Before applying the randomization. After which the levels of randomization of data may be selected 210; randomness may be increased by breaking files apart into randomly sized data blocks 220.

[00328] Honey trap and / or disinformation may be added 230. Then the system may generate random storage locations for the data blocks 240; and proceed to store the data at random storage locations 250; while creating the sequential pointers recording data block locations to an array to provide a reassembly key 260. Then save the pointers array key to a key storage location. Once this is done the original data files may be destroyed 280. It makes sense to destroy the original files, and in some implementations this option may be a preset feature. So that users cannot injure their own security by leaving unprotected copies of files on the host machine.

[00329] The final process step may be to add "*Tally*" data 290 or other bespoke data to the key may be performed at other times in the processing and need not necessarily be done last. But there may be reasons to do this last because it does not necessarily need to be randomized or encrypted.

[00330] Tally data can include things such as the IP address of a device, or historic transaction data and details of previous logins etc. Tally data may provide a useful security check due to the inherent randomness in the minutia of our relatively chaotic lives. These details of minutia events occur in a specific and unique order. But once logged these historic events can be as unique as fingerprints, and details of which users may be able to recall. As will be shown later, an example is provided where Tally data is added to a bank card. Furthermore, it should be noted that Tally data from a mobile phone or other device is also envisaged.

[00331] A lot of the details of hardware that follow are capable of inclusion within a wide and growing variety of devices. The subject technologies are not tied to any particular hardware configuration.

[00332] Figure 25 shows a file 300 comprised of data blocks 1 through 25 which has been stored according to the pointers key 320 into a randomized form 310 that has been written by the read / write head 300 to a local hard disc drive 315. But where, although the pointers array key 320 may be stored to the local hard disc, for increased security it is shown here be stored into a removable USB drive 330.

[00333] The latter is more secure because the USB drive may be isolated in operation from the data and may be removed from the system and used like a physical dongle. So that the computer can go online and even if hacked its data will be incapable of recovery, so long as the USB key is not connected. This makes the security of large amounts of data on a machine lockable, and in a physical way that is very easy for users to see and to check and apply.

[00334] Furthermore, duplicate USB keys may be kept as a backup key precaution in case of media failure or corruption. This is a very good, very secure, and relatively basic implementation of a simple form of this aspect of these subject technologies. A compatible memory card or drive may be similarly used with a mobile phone, and a mobile phone operating over Wi-Fi or Bluetooth may be used with a PC or Mac and vice versa. Other alternatives and variant implementations are also possible.

[00335] A valuable point to note about the pointers key, which is depicted here as arrows. These are like wormholes from any one part of an information technology system to another part of that system. They are a mechanism for pointing to an addressable location within an information technology system. In these subject technologies they may point to physical locations or addresses, like a postal address. Pointers may be used to point to virtualized locations, and / or relative locations or to mask locations rather like using a Post Office Box to provide an anonymous “black box" addressing system.

[00336] These subject technologies use this simple systems architecture to achieve many things. Humans prefer to see ordered systems that they can relate to their experience of the physical world. This has led to a tendency to try to configure computer data processing architecture according to human sensibilities. Whereas, in these subject technologies though it is not a goal per se, computers are allowed to work in ways that may look messy to human eyes. As does the tangle of pointers in this figure.

[00337] However, computers and even AI have no similar concept of tidiness, and they have no issue addressing memory locations in any particular order or format such as arrays and tables or representations of them. Consequently, these subject technologies may ignore some human based sensibilities and let computers work differently than humans may prefer.

[00338] Figure 26 shows a file 300 comprised of data blocks 1 through 25 which has been stored according to the pointers key, to randomized locations 312 within the available space in a storage device or drive 350. This could be on hardware configured as in the previous figure, or it could be different. That detail is not shown, because the main purpose of this figure is to show that using the pointers key, the same or similar file can be randomly dispersed to any available space. This differs from the previous figure also because in the previous figure the data blocks were randomly shuffled out of sequence but written as a contiguous file. Whereas in this example, the storage locations are randomized into storage locations that can have spaces in between them. This is not inherently more secure, when only one file is randomized into available space. But where multiple files are randomly broken into data blocks and then randomly stored into the same overall available space, then they each make the other more secure. This is because there may be no way to identify any data block as belonging to any file.

[00339] So that this randomization into data blocks that are randomly distributed to locations within a common storage space may provide more security for all those files. The storage device may be a hard disk drive partition, or USB drive, or even a cloud storage location. This is not a closed list of storage technologies or devices. Other storage devices, some of which may not exist at the time of filing this application, may be used similarly.

[00340] Gaps between data block storage locations when filled with other randomized data blocks from other files increase the security of all the files. The more files stored this way in a partition or drive the more secure they all are. This effect conforms to the maxim that "*there is strength in numbers*" which really works and is true in these subject technologies. The more files and data that are randomized into a given storage space, the safer they all become. The smaller the needle of the data blocks of any one file becomes relative to the haystack of data blocks of all files.

[00341] Nevertheless, the pointers array key is no less efficient at retrieving files that are co-mingled randomly into a storage space such as a hard disc partition or USB. So that there is no fall in performance of data read access speeds even as the drive approaches being filled to capacity. Data write speeds may suffer modestly increased overheads when seeking an available random location. Especially, as the storage approaches being full to capacity. When data blocks may be allowed to overflow from one random location to another location in order to accommodate a data-block. When it is too large to fit into a single randomly allocated space. The heuristics of a random data block write, and an overflow program capable of filling a drive to capacity are explained further below in relation to Figure 29 and Figure 30.

[00342] Figure 27 shows an even more secure configuration for data storage. In this case the file data blocks 300 numbered 1 through 25, are randomly stored according to the pointers 350 into random locations 314 within five parallel storage locations 375, 380, 385, 390, and 395. These five parallel storage locations may be five partitions or devices attached to one system, or they could be the hard drives of five servers within a cloud data center, or they could be randomly located within five hard drives, of five servers, at five different geographic locations from within thousands of data centers spread randomly around the USA or spread randomly around the world.

[00343] Yet the pointers keys have no calculation overheads and the ability to retrieve and reassemble the data blocks depends mainly on network access speeds and the speeds of data access at the data centers rather than computing power. This may be the most secure example of randomized storage of the file so far illustrated.

[00344] The pointers key can also be kept at a specialist key holder organization to ensure that even the physical data holder cannot reconstruct the data held in their storage facilities into the file. Persons skilled in the art may be able to see that this may be the most secure data protection system ever envisaged; and that if correctly implemented it could be impervious to attempts to steal and reconstruct files. Beyond any previously known system of encryption. But that is still not as secure as the system overall can be made, as will be further illustrated in subsequent figures and descriptions.

[00345] Figure 28 deals with a situation where there may be compatibility issues between an implementation of the subject technologies and the operating system, and / or the disk management system. Say for example it is perfectly possible to randomize a file into the available space on a hard drive of a computer that already has files stored on it according to the rules of its operating system. There is not going to be any problem writing a contiguous randomized file and reading it from such a system. But issues could arise where, for example, the data-blocks from a file are randomized in and among the preexisting data. Again, there is no problem if each data block is capable of being moved around by the disk management system so long as the pointers key is updated accordingly.

[00346] However, there is a risk of data loss if a disk management system were to move data blocks without updating the pointers key. There are some fixes that can be applied, for example such disc optimization functions could be disabled. Another is to add a terminate and stay resident program capable of staying running in the background. Then for this resident program to update the pointers keys to include any movement of affected data when using a local drive, or to create an updated pointers key. So that previous data-block locations can be updated to point to the new locations at the next opportunity. The pointers key may be updated in real time or in two stages. This does not matter per se as long as no attempt to use the affected data-blocks is made until the pointers key has been updated.

[00347] Figure 28 looks at a less optimal commercial outcome, or a time between this application and full market penetration of the subject technologies; and explains how a version of the subject technologies may be safely used with an otherwise incompatible disc management system. Because it is the case that the subject technologies are capable of being implemented without such cooperation and standardization.

[00348] Furthermore, it may well be the case that organizations such as the US military may not be willing to wait for Microsoft and Linux developers and data center operators to catch up. Defense systems that handle sensitive data do not have to wait for the bigger players in the tech sector to begin to benefit from these subject technologies. If they use their own bespoke operating systems for data handling and addressing; then they will probably be able to apply the subject technologies in a way that suits them best. If they need a short-term fix the details of Figure 28 may help such potential early adopters to get these technologies into service sooner. On the systems they have, as they are now.

[00349] The alternatives are arranged from left to right with the most secure options being the ones on the right and the slider 400 visually signals this fact. Alternative process 410 is to not create any secure space or reserved space for randomized data suitable for storage in any locations at all. This can work with caveats. Firstly, a file that has been broken into data blocks and then sequentially randomized can be written contiguously and be dealt with just like a regular file. No problem.

[00350] Secondly, data blocks can be created and randomly stored into non-contiguous disc space locations, that may be interspersed with other files and data blocks from other files. The issue being the need to update the pointers key to these files if a disc management system were to move any data blocks or files. The randomization software may stay running and watching in the background to create a pointer update file capable of updating the pointers key at the next available opportunity. This is also no problem.

[00351] Alternative process 420 is to create a bespoke secure space in local storage, this may typically be an HDD or SSD, but which may also be a removable media such as a USB connected flash drive, or SSD, or even HDD. The good things about this option are that partitions and drives can be hidden, and / or access to them prevented unless it is by compatible software. The downside of using a bespoke secure space within any storage system is that it may provide an identifiable location in which a hacker may know or be able to infer that there is likely to be sensitive information.

[00352] Security may be improved by keeping the pointers key on a separate media which could be a USB drive, or a Blue Tooth accessible device, or local area network or wide area network, such as cloud-based key storage. A USB drive, or mobile phone with a Blue Tooth link may be used like a data access dongle in such configurations. These possibilities are set to continue to grow.

[00353] Alternative process 430 checks for and / or reserves then uses compatible storage space on local area networks, and / or wide area networks, and / or the Cloud in cooperation with a compatible version of the subject technologies and / or a compatible helper application. This may be the way that the technology is able to grow into the available space in the Cloud from where it may be able to displace other storage paradigms. Because there is likely to be little or no extra cost, for this improved security.

[00354] Process step 440 is that whichever higher-level option was selected to run the randomization application, according to selected options. The system can buffer data via a random-access memory disc, or via random access memory or other short-term transitory solutions. Furthermore, there is for discrete portable randomized files an option that may be selected to add start, and end of file markers 450. So that the first alternative is for simple randomized files and key to select and use any storage 460; or the second alternative for randomized storage to select and use secure local and / or remote storage 470. With the process terminating in the storage of data and keys according to selected options 480.

[00355] Figure 29 is a module for use within the storage allocation and data writing operations; the logic is expressed in the form of a block and flow systems diagram. The diagram and its decision-making method, with read and write functions that are capable of implementation may be coded in a desired programming language. Figure 30 is similar. Together these two modules provide the capability for the identification of random storage as being available, and for writing into it the randomly allocated data-blocks.

[00356] Figure 29 addresses the situation thus: to check if a random storage location is available 500; then if available write the data-block to that storage location 530; or if it is not available or less than the minimum usable size, then generate another random storage location 510 and then loop the process back up to step 500 to check if the random storage location is available, and so on. Noting that a feedback loop to the process can provide information as to which locations are used and which are rejected, and store details of the locations used in the pointers key 520.

[00357] This looping data-write process runs on to process the next data block into randomized storage 540 as many times as is needed, until all the file's data blocks have all been randomized into new random storage locations. Creating the pointers key by feedback 520 to the main application 590.

[00358] Figure 30 is similar, but it also has greater ability to cope with data blocks that have sizes that are randomized between minima and maxima; as well as being randomized to disparate random storage locations; and it can also be used in a situation where a drive is getting quite full. Where the spaces available are smaller than the data block for example. This is achieved by allowing the data blocks to overflow from one location to another available location. To achieve this a continuation pointer 565 may be placed at the end of the first fraction of the data block, which points to the start location of the next available overflow location 569.

[00359] An alternative approach is to truncate the data block and write the remainder of the data block to a space with its own pointer key, in the main pointer key. The choice between the two approaches may depend on whether the pointers key is allowed to grow, or the data blocks are allowed to grow in size to accommodate the additional pointers.

[00360] Having explained the alternative process in the abstract, the structure of the exemplary features of Figure 30 are firstly that it is running as a processing module under the main application 590. The first process step for this module is to check to see if the random storage location is large enough for the next data block 550. Then decide if the data needs less than, or an equivalent amount of storage to the available contiguous storage, then write the data 555. But if the data block needs more than the available contiguous storage, then write data to the available space, and allow overflow 560 to a secondary location. By adding a continuation pointer 565 at the end of the primary location that points to the start of the next available location 569; or alternatively add an equivalent pointer into the pointers key that points to the start of the secondary location 569. Then write the remaining data to complete the data block 570. With the pointers key to the location(s) used being fed back to the main application 575. Thereby enabling data to be randomized into all available space in a storage location, drive, or partition.

[00361] This overflow aspect of this program may also be usefully adapted, to compress files to remove, or fill up slack spaces. This may also be used to ameliorate drive fragmentation, without reducing the randomness of the storage allocations. Indeed, it may improve the randomness within contiguously used storage and allow it to be filled to capacity. This may be helpful, because the more randomized data there is stored in a given space, the more it can all help to protect each other. There being strength in numbers (of files) as previously explained.

[00362] Having described and explained the reduction to randomized data stored according to the subject technologies, this specification next addresses the retrieval and reconstruction of the data using the pointers key. This operation may be implemented in a modular way under the control of the main application.

[00363] Figure 31 shows the operations running within a main application 690; where the user selects files to work with 600. This causes the system to retrieve the pointers key to storage locations from key storage 610. The detail of where this key storage is, whether it is a local USB drive, or a distant Cloud key storage location is not shown. So that the common elements across many implementations of the subject technologies can be shown more clearly by not obscuring the basics with those details.

[00364] The pointers key is used to interrogate the storage locations of the locked file 640 to which it points so that the data-blocks those locations contain may be retrieved and reassembled back into the order of the original file 650 by recreating the sequence in accord with the sequence of the pointers key to provide a reconstructed open file to work with 660.

[00365] Figure 32 describes some more complex data security examples where pure randomization can be integrated with other aspects of the subject technologies including the use of Tally data, and / or honey trap data, and / or the subterfuge of using encryption on the files to be protected in addition to randomization. The last being useful, for concealing the true nature of the logical randomization of the data. To further bamboozle and befuddle would be hackers, and crackers.

[00366] Figure 32 is essentially the reverse of the process and steps including the hybrid aspects of the subject technologies shown as being used to protect data in Figure 24. In Figure 32 the first processing step may be to check the IP address and / or device and / or other bespoke Tally data if any are available 700. The pointers key is retrieved 710, then using the pointers, data is retrieved from the locations pointed to 720. If any honey trap or disinformation data was included among the stored data this may be stripped out 730; and if there was any post randomization encryption as may be useful in the case of a contiguously stored but internally randomized file, then this encryption can be reversed according to whatever its rules are (not shown) 740. So that the randomized data can then be reassembled back into its original form by the process step of reassembling the data according to the pointers key sequence 750. The further step to then reverse any pre-randomization encryption 760 may be executed. So that the reconstructed data is finally restored to its original form pure of any encryption whatsoever, and the data and / or its files are made available by the application for use or editing by the user 770.

[00367] Users may be provided the function to save edited data and / or files back and updating all 780 probably according to their preset or other default options (not shown but likely to be like the process illustrated in Figure 24 again). The final process step being to generate and save a new pointers sequential storage key as the data is re-saved 790, according to the subject technologies. In subsequent figures just as the detail of the operations of Figure 24 were expanded upon subsequently, this specification now similarly expands upon these operations in the figures that follow.

[00368] Figure 33 shows at the hardware level randomized data blocks 1 – 25, being retrieved from where they were stored in Figure 27. With the five disparate storage locations 830 containing data blocks 5, 14, 21, and 2, 840 containing data blocks 4, 24, 11, and 9 etc., with locations 850, 860 and 870 containing the rest of the data blocks that were randomly stored within those locations. The pointers key is used by the main application running on a system according to these subject technologies, which operates on them to reassemble them back into their original form into the file 810 on the right of the figure.

[00369] Pointers may connect data at great distances or locally to a local machine, the physical distances of which are virtualized here. Pointers not only have the benefit of being uncomplicated to use, but there is also no math or calculation overhead to doing so, they are what they are. They do not consume processor resources the same way that cryptography does as it crunches its way through the math, nor do they require processing power to determine recovery locations in data access operations as some hashing functions may.

[00370] Figure 34 shows the same file comprised of data blocks 1 – 25 that was recovered from randomized storage in Figure 33, and as explained in Figure 32, except that this file 900 has been worked on, so that data blocks 3M, 11M, 12M, 13M, 17M, 19M and 21M have all been modified. The modified file is again randomly distributed to five storage locations 930, 940, 950, 960 and 970. These may be separate server hard drives, in separate data centers within the Cloud infrastructure (135 of Figure 19, Figure 19A, and 19B) which Cloud could have hundreds or even thousands of data centers. The pointers key may be stored also in a Cloud location or locally or to removable media (130 of Figure 19A, and Figure 19B). The options are multiple. So that with this figure it has been demonstrated how data may be randomized, stored, retrieved, then stored again according to these randomizing subject technologies.

[00371] All security technologies have weak points, and one such weak point is where a local copy of a file may remain assembled in the storage of the device where it has been worked upon, or where it has been stored. Software exists that can recover deleted files, and data recovery experts and forensic scientists can even recover magnetic impressions of data that has been deleted and even data that has been deleted and overwritten. The Department of Defense has specified ten overwrites as a standard for the secure destruction of magnetic data on hard disk drives. But it is likely that even this is not fully effective against the very best laboratories as may be used by spy agencies.

[00372] Ideally, once data is randomized into disparate storage locations within the cloud, arguably it should not be reassembled into a local copy on a local hard drive if it is highly secret. Rather it should ideally be worked on across the cloud infrastructure. To avoid creating a local copy in storage that could be searched for and recovered.

[00373] Furthermore, it may be held locally, but only in random access memory (RAM) so as not to leave any copies of any part of the file, on a local hard disc drive (HDD). As often occurs due to the operating system buffering to a part of the HDD that has been reserved for buffering to work like a virtual RAM-disc; or page file.

[00374] Keeping the data entirely in real RAM is now much more practical than it used to be. This is because the cost of RAM has fallen, and its stability and size have increased in parallel with improving transistor processing units. This trend will likely continue, along with improvements in processors, according to Moore's Law.

[00375] However, where a virtual RAM-disc or disc-based page file is used, then the data it contained although transient in nature may nevertheless leave a trace copy of itself that also ideally needs to be securely deleted. Especially where security needs are high. Such as in the intelligence community, military and defense contracting organizations like Lockheed Martin, RTX, and Boeing etc.

[00376] In data security critical operations, whether locally or remotely in the cloud there is a need for secure deletion of files including any RAM-disc or page file data. Using these subject technologies, it may be possible to exceed current DoD specifications.

[00377] Figure 35 provides a secure deletion process. Which is capable of operations as a module running under the main application of these subject technologies on a local or remote machine, or both or even as a standalone program. The secure deletion application is running on a machine under a main application.

[00378] Randomized data held purely in cloud storage is much less of a risk. Provided that is it stored in randomized locations. Then all that is really needed in most cases may be to securely delete it – is the secure deletion of the pointers key file; and release back to the system of the storage allocations to which it pointed. So that in the following description, it is helpful to include an array of pointers or key file in the types of files that this module or sub-program can delete.

[00379] The first operation 1000 is to randomize the file name string 1010 and extension

1020. This is accomplished by repeatedly renaming the file with random or a succession of fictitious names 1010 that are at least as long as the original name string, then to do the same thing over-again to the extension or file-type descriptor 1020 which shreds these attributes, and to do it as many times as required. The DoD standard of ten overwrites may be exceeded if so desired.

[00380] The next process step 1030 is for the randomized overwriting with random data ("*0* - *n*") from zero times to "*n*" times, of the data stored at the file locations to which the resulting name string 1040 and extension 1050 relates. The reason why it may be desirable in some cases to set the number of overwrites to zero in relation to the data is because, if it is already deep within a cloud infrastructure, and already broken up into data blocks that have been distributed into random storage locations. Once the file name and file extension have been removed, then freeing up those randomized locations may be all that is needed. Because it would be practically impossible to reconstruct the file once the file name no longer matches any pointers-based key. This approach may also be preferred within Cloud-based infrastructure because there is no need to waste resources overwriting such irrecoverable data. The ability to set this parameter to zero may thus save a considerable amount of unnecessary processor use and write operations. That may have the further benefit of reducing wear on the data storage media and saving on power consumption .

[00381] Noting that the DoD specification for the deletion of data is ten overwrites also noting that in these subject technologies, this is not necessarily accomplished by simple repeated deletions with the zero character or one value. So that if this becomes a DoD compliance issue then the option to increase the number of overwrites may still be achieved by setting the number of overwrites to meet the required standard. This is more likely to matter at the local device level, if that device or storage media might fall into hostile hands, for example on a battlefield system.

[00382] Use of random data for the data overwrites may provide an improved overwriting process of randomized, or pseudo-random overwriting of the physical data according to these subject technologies. So that it should be more difficult to distinguish the underlying data that is being erased from every level. Because probably the data will be overwritten in a binary system on average around about fifty percent of the time with the same value that was there before. This comprises a more sophisticated method of random or pseudo-random overwriting. Being probably harder to back engineer than simple repeated deletion and overwriting with a predictable value such as zero or one.

[00383] Though not shown the security, and trade-craft disinformation factor can be further boosted by being overwritten with disinformation data that is comprehensible and not random. So that if a really clever forensics team or AI program does find a pattern hidden in one layer of recovered deleted data, it could be comprised of intercepted communications data files, or maybe some plausible disinformation between delete cycles. Disinformation that may be of keen interest to an adversary if true. So as to put a helpful distraction data pattern in there.

[00384] After which is shredding by overwriting, the next operation 1060 is to delete the data, and file name by releasing its now vacant storage locations back to the system(s) local and remote (if applicable) for re-use 1066. This may be accomplished in a Windows system for example by removing the file entries from the registry. Re-use and more overwriting with other files as may then occur in this context is also a good and helpful thing.

[00385] Images can contain secret information of many forms, and they can also be processed and stored according to the subject technologies. This is true for secret spy satellite photographs of adversaries' or allies' weapons systems, that it is desirable do not spread beyond the people who have a need to know their details. Other images can require protection for personal reasons.

[00386] Figure 36 is an image file randomization 1100 that uses the pattern of a randomized spider's web comprised of a variable swirl 1110, and a variable pattern of spokes derived from patterns that may be randomly varied within maxima and minima to cut or break up an image into data blocks 110. That are then capable of being randomly allocated into storage locations; and to also produce a pointers array key in accord with the subject technologies.

[00387] Figure 37 is an image randomization 1140 that uses a pattern of vertical lines 1150 and horizontal lines 1170 to create a pattern of overlaid rectangles capable of dividing the image into data blocks 1150 that are randomly sized between preset minima and maxima, and which minima and maxima can be varied or randomized within an identified zone of highest secrecy. To form smaller data blocks for random storage as needed 1155. The image being thus cut or broken up into larger data blocks 1150 and smaller data blocks 1155. That are then both capable of being randomly allocated into storage locations; and to also produce a pointers array key in accord with the subject technologies.

[00388] Figure 38 is an image randomization 1180 that uses a jigsaw pattern randomized to produce slightly surreal contours 1195 to create a pattern capable of dividing the image into data blocks 1190 that are randomly sized between preset minima and maxima, and which minima and maxima can be varied, and reduced within an identified zone of highest secrecy (not shown here). That are then in all their variable shapes and sizes capable of being randomly allocated into storage locations; and to also produce a pointers array key in accord with the subject technologies.

[00389] Once created these cut patterns can be used similarly on multiple images, and they can be selected randomly from a selection of cut patterns. Or they can be created on the fly as bespoke cut patterns. Probably, users of mobile phones might like to design their own cut patterns by doodling them onto the screens of their devices. Next the systems block and flow diagram capable of performing these tasks is described.

[00390] Various components and blocks may be arranged differently, for example in a different order or partitioned in a different way; all without departing from the scope of the subject technology.

[00391] Figure 39 is a modular operation, running under a main application in accord with the subject technologies. Firstly, the image cut method is selected to create the data blocks 1200. After which comes one of the alternative process steps: 1210 provides simple cut grid patterns, which can be as small as one pixel, and variable within maxima and minima in the form of “x” pixels by “y” pixels; 1220 provides jigsaw type cut patterns, generated with random variables and scale options; 1230 uses scribbled cut patterns provided by users and that are imposed onto the image; and 1240 is a catch all alternative option for any other suitable cut patterns that are possible. The full range of which is potentially infinite and not further expressed for that reason.

[00392] The next processing step is to supply these variably sized data blocks for randomized onward routing to their randomly allocated storage locations 1250. As visually drawn in perspective as the stream of data blocks of varying sizes 1260 is being output. These data blocks being provided to other parts of the main application (not shown) and / or to helper applications at their ultimate storage locations.

[00393] Figure 40 is similar to the previous figure, except that module related to processing still images, and this module relates to streaming media which includes a succession of images. This module may be run within or called to run from within a main application. The first step of which is to select the movie, or live feed and cut method to create data blocks or stripes.

[00394] Then comes the alternative steps: 1310 to cut individual movie frames up using cut patterns which would be very secure; or 1320 to cut between frames; or 1330 cut to variable randomized or fixed numbers of frames within maximum and minimum parameters; or cut to variable randomized or fixed timed lengths, between minimum and maximum parameters, such as a 1 minute minimum and a 15-minute maximum. Then to supply these variable or fixed data blocks or stripes for onward routing to randomly allocated storage 1350. So that the stream of data blocks or stripes of fixed or variable sizes 1360 is output as such and can then proceed to be written into randomized storage locations according to the subject technologies.

[00395] Figure 41 shows block systems and flow diagram aspects in the context of a schematic representation of the upload that may be from a computing device 111 or 113 of randomized data blocks (or stripes) to randomized locations within the cloud, as a process or module operating on cloud infrastructure.

[00396] According also to the systems shown in Figure 19 and Figure 19B, which may (optionally) be configured by methods and with apparatus to include the dual channels isolation 14 aspects of DICE between a user device 111 or 113 which may operate similarly to the server side apparatus 104 in the Cloud infrastructure 135 shown in those figures.

[00397] The stream of data blocks (or stripes) coming to the cloud 1470 is coming from another module or process running a process that may be like those of Figure 19, and Figure 19B as described in more detail in the materials relating to the figures up to and through Figure 43. According to these subject technologies on a user device 111 or 113, over a network and arriving over the cloud-based internal communications channel "X" 152 as the data blocks (or stripes) 1470. These are being uploaded to randomized locations 1480, and 1490 and within the cloud infrastructure locations marked "X" in the cloud data centers 1430. These uploads may occur in parallel, for extra speed where the available network infrastructure permits.

[00398] Pointers directing and / or recording where each data block (or stripe) is stored, are sequentially stored in an array 1410 in a separate cloud infrastructure 1400 as the upload proceeds. The separate cloud infrastructure 1400 which may be isolated 14 from the cloud infrastructure 1430 where the data blocks may be randomly stored marked “X”. The pointers and data being stored in real time within the process time frame snapshot 1450 are kept in sequence so that the key may be comprised of the sequence of pointers 1410 as it corresponds to the locations of the data blocks (or stripes) that is uploaded to the separate and isolated cloud key storage location 1400, and that may be backed-up to, or mirrored in alternative storage 1420. To avoid loss of the pointers key in the event of a failure of the primary Cloud based key 1410 storage. A similar alternative mirror or backup location is also shown for the cloud-based data storage 1460.

[00399] The curved arrows are used to show data paths to and from their storage locations marked “X” 1480, and 1490, and the creation and storage of the pointers to the locations marked “X” being recorded within a sequential array structure 1410, and its upload to and storage in a separate and isolated cloud key storage location 1400 as a key within that cloud.

[00400] Uploading and recording of the sequence of pointers keys 1410 may be accomplished via a separate channel “Y” 153, which may use a separate thread, or port or isolated network connection according to the DICE subject technologies operating through a networking card and / or separate communications network (that is not bridged) and / or routing that however implemented may be (and for maximum security should ideally be) separate and unbridged to provide isolation 14 from the channel “X” 152 that may be used for data.

[00401] The cloud storage locations 1430 that may each be within data centers, are shown as a receding line of five clouds, and it should be remembered that there could be hundreds or thousands of these. Though there is not enough space to show more than a few. In the same way and for the same reason as only a few data blocks (or stripes) are illustrated in this snapshot correlating to the time frame window 1450. This is because the illustration would be obscured by detail if more data blocks (or stripes) and pointers were shown passing over a wider time frame.

[00402] Having addressed an upload that is compatible with embodiments of Figure 19 and / or Figure 19B. Which may be tailored to suit users who want or need more local control of their data.

[00403] We next consider a similar but technically different mass market solution that is tailored more closely to the needs and preferences of regular consumers, and large Internet service providers and operators of Cloud storage solutions. Such as AWS, Google, and Microsoft as well as applications such as online banking and medical services, along with social media services such as Facebook. Which uploads are more likely to be suited the embodiment provided in Figure 19A, and / or Figure 19B. Which are also intended to be compatible with the next Figure 41A. Furthermore these embodiments may be used with DICE. Full compliance with which is desirable but is not essential to the DSURF subject technologies. Figure 2 provides an example of the DICE and DSURF technologies being used together in a peer-to-peer transfer of files secured by DSURF over networks secured by DICE which allows the encryption keys to remain isolated from the files in transit.

[00404] The embodiment provided by Figure 41A, may be used like a black box module 1497, residing in a Cloud based network infrastructure, such as a data center or collection of data centers. On the outside of the black box, client devices 1496 may connect to system 1497. Systems which may be compatible with this configuration are provided in Figure 19A, and Figure 19B. Using a website web page or web-application 106, running on a user device 113. That may access and cooperate with a system such as this as a black box1497 via communication channel 1 and / or channel 2.. Into a server-side network 104 of a Cloud infrastructure 135, coupled with a service.

[00405] In these embodiments the client devices 1496 may operate communication channel 1 and / or channel 2 according to the current art, and / or they may operate according to the DICE subject technologies. The externally connected user devices 1496 are not shown within this embodiment 1497 of a cloud based black box embodiment. Wherein these subject technologies may operate as follows.

[00406] Figure 41A addresses uploads including file processing; and Figure 42A addresses downloads including file processing. These being the two halves of the operation of this embodiment 1497. Which is an embodiment of these subject technologies that have been tailored to the needs of regular consumer users who may be using Cloud storage for their data. Which embodiment is capable of providing DSURF to them for use which may be operated by service providers. Requiring little more than to download and install an application or browser extension, or to use a secure website on their connected client devices 1496 and this cloud-based service. The internal workings of which are not apparent from the outside.

[00407] The user experience of which may be made indistinguishable from the cloud drives and other online services in the art, within a cloud infrastructure. That is capable via these subject technologies to reduce the incidence of large-scale data incursions and theft from big-data holders, and other custodians of sensitive data including financial and medical records and processing.

[00408] File processing and handling is controlled by one or more controller servers 151. Files received from connected 1496 client devices 111 or 113 may be processed and broken apart into data blocks, and these data blocks may be similarly sized, or randomly sized between a minimum and maximum 1441 as can be seen with data blocks “K”, “L”, “M”, 1471. Then randomly stored by uploading the data blocks into unique random storage 1441 locations marked “X” 1480 and 1490 etc., within a cloud-based network infrastructure 1430. These uploads may occur in parallel, for extra speed where the available network infrastructure permits.

[00409] The cloud-based infrastructure being comprised of a plurality of available storage locations and / or devices, within a data center, that may be part of a plurality of data centers. The greater the scale, arguably the better for data security, and for data access speeds. Which may be very fast, or even close to instantaneous because these storage locations may also be accessible in parallel. Which may provide faster data upload and download times than is possible via serial storage to a single location. So that data storage, according to this embodiment, using parallel processing into randomized storage locations may have two considerable benefits of (i) improved speed, as well as (ii) improved security.

[00410] Data uploads into randomized cloud storage locations 1430 of the data blocks 1471 may be via one or more secure channels “X” 152, these secure channels may be isolated from the secure channels “Y” 153 that may be used for the uploading pointers array-based keys which pointers point to the locations of the data blocks in storage. So that pointer “K” identifies the location of data block “K” marked “X” in location 1490; and pointer “L” identifies the location of data block “L” marked “X” in location 1480, within the cloud infrastructure 1430. Furthermore, the pointers array keys may be stored into a separate cloud storage infrastructure 1400, locations 1410; and which may also be separate, isolated 14 and distinct, from the cloud storage infrastructure 1430. Another way of thinking about this is that the DSURF black box is operating internally according to some of the same, or similar networking processes of operation as DICE may use to provide the dual isolated 14 channels X 152 and Y153. Which may or may not be operating outside the black box to connect it with client devices 1496.

[00411] Thanks to improving network bandwidth, and access speeds the fact that these subject technologies may use parallel processing for the upload and download of data, such embodiments of these subject technologies may be very fast, as well as being more secure.

[00412] Persons skilled in the art will appreciate that the alternative or mirrors or backups storage for data 1460, and for storage of pointers keys 1420, may be held within their own isolated 14 locations similarly to their live and active equivalent versions. The use of backups and mirrors is an element of cloud infrastructure.

[00413] Next this description looks at downloads according to these subject technologies in a slightly differently configured embodiment, before returning to further consideration of this embodiment being run in reverse for file reconstruction, file processing and downloads.

[00414] The download shown in Figure 42 may be compatible with use according to the embodiments of Figure 19, Figure 19B and Figure 43.

[00415] The encryption security of Apple's mobile phones have shown there is a market for secure devices, that have been difficult even for the FBI to decrypt. It is probably the case that data encrypted according to these subject technologies may be even harder to decrypt. So perhaps Apple and their users may prefer such systems according to these subject technologies, as the encryption used in their current products may eventually become obsolesced by quantum decryption technologies such described in relation to the process provided by Figure 47.

[00416] Figure 42 shows block systems and flow diagram aspects in the context of a schematic representation of the download 1442 that may be to a computing device 111 or 113 of randomized data blocks (or stripes) from randomized locations within the cloud, as a process or module operating on cloud infrastructure and may be in cooperation with and / or under the control of a user device system 111 or 113. This may have the same component parts and configuration of infrastructure as in the previous Figure 41 showing the upload that preceded this download, and reassembly of the data blocks (or stripes). That can be seen to be operating rather like a zipper as the pointers are applied to call down and used to zip back together the data blocks (or stripes) 1442 into a reconstructed copy of the original file or data stream 1475.

[00417] This is an important aspect of the subject technologies. Because it may form the basis for defense cloud data protection; as well as for civilian data storage and / or streaming media applications where intellectual property rights are being protected, and monetized. As such it may be one of the more commercially important aspects of the subject technologies.

[00418] The downloading and recombination operation may be operated by the client device 111 or 113 downloading the pointers key 1410 from the cloud key storage 1400, or from the alternative mirror or backup key storage 1420; then reading sequentially through the pointers key to identify and call downloads of the data blocks (or stripes) contained within the storage locations of the data blocks (or stripes) at locations marked “X” within the cloud data centers 1430, identified here in this snapshot as data block 42 which tallies with pointer 42 and is called out as item 1490, and data block 43 which tallies with pointer 43 and is called out as item 1480; or if problems are encountered these data blocks (or stripes) may be downloaded from the alternative other or mirror or backup data storage 1460. After which these data blocks (or stripes) are recombined 1442 or zipped back into their original sequence to provide a copy of the original file or data stream 1475. This aspect is specifically addressed in relation to the embodiment provided by Figure 43 in the context of data streaming.

[00419] The separate cloud infrastructure used for storage and retrieval of the pointers array based keys 1400 may be isolated 14 from the cloud infrastructure 1430 where the data blocks may be randomly stored marked “X”. Downloading, the sequence of pointers keys 1410 may be accomplished via a separate channel “Y” 153, which may use a separate thread, or port or isolated network connection operating through a networking card and / or separate communications network (that is not bridged) and / or routing that however implemented may be (and for maximum security should ideally be) isolated from the channel “X” 152 that may be used for data.

[00420] The data storage locations being located in a similarly separate and isolated cloud infrastructure 1430, via a similar separate and isolated channel “X” 152. For higher security applications these can be implemented as non-connected or non- bridged network cards that may connect to different networks or routes such as through a cell phone network in one channel and a home or business Wi-Fi channel configured according to the DICE subject technologies. Some users may not need to use hardwired channel isolation measures. For them, using separate ports and threads for their channels may suffice. At least until the age of quantum decryption risks may force them to use more secure configurations also operating according to the DICE subject technologies. The next Figure 42A is addressed in particular to the more likely demand for a black box implementation of these subject technologies, that may be provided via a website or browser extension or other application.

[00421] Figure 42A addresses downloads including file processing via the operation of this embodiment as a black box1497. Which is tailored to the needs and preferences of most regular consumer users, and big data service providers. Requiring little more than to download and install an application or a browser extension or to use a secure website via a connected 1496 client device. That is capable of reducing the incidence of large-scale data incursions and data theft from big-data holders, and custodians of sensitive data.

[00422] File processing and handling is controlled by one or more controller servers 151. Data downloads may be called by a connected client device 1496. Example user client devices may be the workstations of a social media operator, or financial or medical service provider, or a cloud drive user. Data blocks that were previously randomly sized and then randomly stored into random cloud storage locations such as the data blocks 1471 of Figure 41A may be downloaded by running the process 1497 in reverse to provide the data blocks "K" and "L" 1476 being downloaded and recombined into copies of their files 1443. Which downloads may be via one or more secure channels "X" 152. Which secure channels may be isolated from the secure channels "Y" 153 that may be used for downloading the pointers array-based keys which pointers point to the locations of the data block storage.

[00423] Pointer "K" identifies the location of data block "K" marked "X" in location 1490; and pointer "L" identifies the location of data block "L" marked "X" in location 1480 within the cloud infrastructure 1430. Furthermore, the pointers array keys may be kept in and downloaded from a separate cloud storage infrastructure 1400, at locations 1410; and which may also be isolated 14 from and distinct from the cloud storage infrastructure 1430.

[00424] For example, the cloud storage infrastructure may be owned and operated by a bulk provider, and the keys may be stored separately and isolated by the bulk storage provider or in separately owned and operated specialist key storage facilities. So that potentially not even the host of the data would have the information from which it may be reconstructed back into copies of the original user files. The isolation of data storage and communications channels "X" 152 from key storage and communications channels "Y" 153, should suffice to make any data theft practically useless to hackers. Because if the data blocks are not reconstructed according to the pointers array keys, then there is no presently known way to determine how to reconstruct copies of those files. This may provide a step change improvement in the battle against big data theft!

[00425] Thanks to improving networking bandwidth, and the fact that these subject technologies may use parallel processing for the upload and download of data, such embodiments of these subject technologies may be very fast, probably faster than the prior art using serial processing. As well as being more secure. This is also because uploads and downloads of the data blocks may occur in parallel where the available network infrastructure permits.

[00426] Figure 43 shows process, running under a main application that may be operated by a smart phone, smart TV, Tablet or PC. It could even be running on the US military version of the Internet and relate to classified material being shared between the senior defense staff on the East and West coasts, as well as officers in the field – as part of a classified teleconference. Wherein, a data block (or stripe) stream from either satellite and / or antenna and / or dongle and / or a wide area network (such as the Internet) and / or telephone line and / or other data or communications channel 1500 provides data blocks (or stripes) to the device(s) upon which this module is running.

[00427] The data is in the form of blocks (or stripes) of media data 1510 that is 2.96 minutes' worth of randomized data stream; 1520 that is 9.82 minutes' worth of randomized data stream; and 1530 that is 3.45 minutes' worth of randomized data stream. All of which may travel different routes over networks from disparate randomized storage locations. Though that level of detail is illustrated in previous figures but not shown here for the sake of clarity.

[00428] What enables the stream of randomized data blocks (or stripes) to be downloaded from their randomized storage locations is the data block randomization keys from satellite and / or antenna and / or dongle and / or the Internet (or similar wide area network) and / or telephone line and / or other communications channel 1505. These data randomization keys may be comprised of arrays each corresponding to a randomized data object and which keys are 1515, 1525, and 1535 provide pointers to the download address or storage location of each of the respective data blocks (or stripes) in the same order 1510, 1520, and 1530.

[00429] The DICE subject technologies may add more security in transit over networks, which may include the Internet by isolating the data travelling through a channel from the encryption and / or pointers-based key that may be shared over a different and separate channel. Though this is not essential to run this streaming media embodiment from DSURF-based storage within a Cloud infrastructure. Indeed, where a channel is based on optical fiber or in space applications point to point security in transit may be boosted by quantum entanglement-based protection.

[00430] Furthermore, the one-to-one relationship between data blocks or stripes and their addressing call down pointer as well as being used in sequence to call down the data blocks for reassembly, may contain additional complexity not shown here where the frames of movies may be randomized by cut patterns and / or randomized out of sequential order, and these changes may also be recorded in arrays of pointers. Nested levels of pointers that may carry this additional randomization aspect data may be carried within the keys as parallel arrays, nested arrays or even an array of database files. Capable of providing very secured streaming media!

[00431] However, this figure is capable of illustrating not just what it is doing but also that a data streaming service can be run using the subject technologies of DSURF, to provide previously unattainable levels of security that may be further enhanced by using DICE in combination with DSURF and / or other technologies; and that from the user device perspective this may be implemented as two data streams or threads running concurrently. Provided the bandwidth to read ahead of the images being viewed is present. With the result that users may enjoy high speed playback and security of the information in the streaming media.

[00432] Where security really matters, the two streams of data blocks (or stripes) may for improved security travel over physically different channels of different infrastructures. So that the pointers key 1505 may be provided by a coupled mobile phone perhaps using Blue Tooth and using its cellular service provider network, and the data blocks (or stripes) stream 1500 may travel over the Internet. Many other combinations and channels are possible for use with this aspect of the subject technologies, including using storage media sent by post to provide one or other stream. There are many possible combinations and ways to ensure that both pointers-based keys and data-block (or stripe) streams do not arrive through the same fiber-optical or radio frequency channel and / or transceiver systems. This simple measure of physical channel separation and isolation of channels and / or storage per se may drastically reduce the risk of successful interception / or theft from storage and decryption of secret materials even as against quantum decryption risks. Such as that provided by process for the operation of a suitably programmed binary artificial intelligence and quantum system (BAIQS) hybrid computer similar to the embodiment provided in Figure 47.

[00433] As in previous figures the pointers array-based keys may be used to reconstruct the files at the server-side level of processing. Indeed, in black box type operations that may suffice for many consumer applications on regular devices. Or as in this embodiment of process 1588 the pointers array-based keys may be called down then used to call down the blocks or stripes of data to be recombined into copies of the original files, or data stream at the level of the user device. Which may provide the security levels desired for sensitive streaming media, as may be desirable in multi-way teleconferencing applications as the military and intelligence services may need between distant locations, and as part of joint all domain command and control operations, data analysis and mission planning.

[00434] Figure 44 represents the current state of the art, for a banking card 2600 with a trading name Duff Bank, which contains a silicon chip, transistor, and memory system 2610.

[00435] Figure 45 however, is a banking card 2620 with a trading name of Random Bank which incorporates several aspects of the subject technologies. There is in addition to a chip transistor and memory system 2630, a connection 2635 comprising an interface capability with an additional memory device 2640, capable of storing inter alia randomized and hence secure data, that may include a pointers-based key or details of encryption and / or tally data and / or any other helpful data 2650 in excess of what is used in the current state of the art. One or more of these, or any usable size and shape may be included on a banking card, though only this one example 2640 is shown. The additional memory is connected to the first chip to allow it to interface through its contacts with existing chip and card readers.

[00436] Furthermore, there is a photon-based storage zone capability 2660, wherein light can be used to write and / or to read data, 2660 capable of storing data that may be randomized secure data, that may include tally data and / or any other helpful data. This would require changing the current chip and card reading systems and is thus probably less attractive to banks and credit card companies than the additional memory module option 2640.

[00437] Tally data, in any case – however recorded and retrieved – may include data relating to user activities, usage patterns, bio-metric data, and / or any other helpful data. Helpful data may also include historic shopping data and as well as serving the cause of security, may subject to applicable laws in a jurisdiction be used to trigger the presentation of information including to a user, via an interacting device or system. Such as a mobile phone or personal computer.

[00438] Furthermore, in jurisdictions where the use of banking cards has been substantially overtaken by the lack of infrastructure for banking card use, and where this is supplanted by using mobile phones to facilitate payment for transactions, then the capabilities may alternatively be delivered within and / or through such a smart phone or other device instead; and / or for home-based users the applicable device may be a personal computer alone, and / or with a card reading capability, or equivalent Internet banking capability. That may furthermore be used to update a banking card linked to the account used to be updated, at the next possible opportunity. As well as networked services and records.

[00439] Figure 46 shows a laser read and / or write module, under the control of a system that is not shown. The local read / write control system 2695 of this device includes a feedback loop 2690 for reading back data and checking data has been correctly written. The feedback loop may also be used for set-up of alignment and adjustment of the device. The local control system 2695, controls the laser read / write head 2680, and the beams it produces 2686 for writing and reading data, to a photosensitive medium 2660 that may be incorporated into a bank card; where it is used to store user data, which may include historic tally data, transaction, and geographic information as well as the more usual bank account data. There is no reason why bank cards could not incorporate and operate an optical storage media capability. However, because the ability to write large amounts of data to small spaces, in Figure 46 the area used for this purpose 2660 is relatively modest.

[00440] Figure 47 provides details of a binary artificial intelligence and quantum-computing (BAIQ) decryption process. Which BAIQ process may receive encrypted data 55 and then apply its artificial intelligence (AI) and / or machine learning (ML) codebreaking expert systems which may analyze the encrypted data 56. Then proceed to attempt to identify data patterns and / or known cribs or clues about the encryption of the data 57. The AI may then run best fit quantum processes looking for prime numbers, factors, products, and sums that may be applicable to the encrypted data 58. The results of which analysis may be presented along with options 59 by a user interface 60. A human and / or (AI / ML including expert system logic) machine intelligence (MI) user may then select from available options 61. Then decide, if the decryption is successful the process may terminate this decryption process 62 or determine that the encryption is probably quantum-decryption-proof or quantum-decryption-resistant. In which case the system may switch to (AI / ML including expert system logic) MI powered codebreaking expert systems and methods 63. Next conducting a deep scan of the encrypted data for hints or evidence that may allow inferences to be made suggesting the use of known ciphers 64.

[00441] Known ciphers include ciphers that may be quantum proof. This is where the "*intelligence*" of (AI / ML including expert system logic) collectively MI may provide the boost of insight that may be logical and may be intuitive but still be sufficiently random that it cannot be solved using mathematics alone. This machine intelligence may thus be able to postulate unknown ciphers, and solutions 65. Doing so with the care and skill of a mathematical genius and an intuitive creative genius at the same time. To operate like a virtual Dr. Alan Turing and quantum computer hybrid. That most humans would be incapable of matching.

[00442] Furthermore, a BAIQS machine intelligence capability will have available to it the ability to leverage and test a large Quantum Model, and / or use true Quantum computations – one or other of which or these in combination with the MI may solve some aspects of a hybrid mathematics and randomized encryption that may include logical ciphers and randomization according to the subject technologies and / or ciphers in the pre-existing state of the art. The great strength of the BAIQS capability in this regard is the ability for the MI to relentlessly keep trying all the different methods and combinations it can think of, without suffering from boredom or fatigue and be able do this at a speed humans cannot match. At the end of which the BAIQS processing may try the best fit approaches 66.

[00443] At any of these stages 64, 65, and 66, options may be presented 59 by the interface 60, and decisions made by a human and / or MI user as to whether the decryption has been successful and what steps of this process to invoke next. In either event but typically upon a successful decryption, the decrypted data may be saved into an appropriate format. Such as a text file, or image or movie formatted file etc., and the process may terminate 62.

[00444] BAIQS capabilities may thus provide the ultimate codebreaking solution that is unlikely to be surpassed for a very long time. It is likely that America's NSA and the UK's GCHQ will want their own BAIQS, and that rival intelligence agencies and militaries around the world will seek to replicate a BAIQS.

[00445] Furthermore, warehoused encrypted data that may have been stored over recent years while intelligence and law enforcement agencies were waiting for such capabilities to become available may provide a boon for those agencies. Which may be able to leverage BAIQS' enhanced decryption capabilities to break the decryption used by adversaries, cyber criminals and fraudsters. From times when they may have believed their encryption was unbreakable. In what has been dubbed by media as the "*Quantum Apocalypse*" in which all the old mathematics-based encryption (such as RSA which has public keys) appears set to become obsolete due to impact quantum computing on the field of decryption and code-breaking.

[00446] *Computing and Hardware Elements*

[00447] Some aspects of the method described above use computing elements that operate software providing functionality to the hardware. The computing device elements may include general purpose computing devices which may include, but are not limited to, one or more processors or processing units, a system memory, and a bus that couples various system components including the system memory to the processor as is known in the art of computers. The computing elements may be described in the general context of computer system executable instructions, such as program modules, being executed by a computer system. In some aspects, some of the computing functions may be provided off device (remotely) by a server (for example, the processes shown in the figures) which may be a cloud computing node connected to a cloud computing network (not shown) and practiced in distributed cloud computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed cloud computing environment, program modules may be located in both local and remote computer system storage media including memory storage devices.

[00448] The computing elements may typically include a variety of computer system readable media. Such media includes non-transitory, volatile and non-volatile media, removable and non-removable media. The system memory could include one or more computer system readable media in the form of volatile memory, such as a random access memory (RAM) and/or a cache memory. The system memory may include at least one program product having a set (e.g., at least one) of program modules that are configured to carry out the functions of embodiments of the subject technology. The program product/utility, having a set (at least one) of program modules, may be stored in the system memory, as well as an operating system, one or more application programs, other program modules, and program data. Each of the operating system, one or more application programs, other program modules, and program data or some combination thereof, may include an implementation of a networking environment so that the device may perform some of the environmental detection and contextual awareness through remote computing devices. The program modules generally carry out the functions and/or methodologies for detecting environmental objects and generating contextual awareness output to the user.

[00449] As will be appreciated by one skilled in the art, aspects of the disclosed invention may be embodied as a system, method or process, or computer program product. Accordingly, aspects of the disclosed invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module," or "system." Furthermore, aspects of the disclosed subject technology may take the form of a computer program product embodied in one or more computer readable media having computer readable program code embodied thereon.

[00450] Aspects of the disclosed invention are described above with reference to block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the invention. It will be understood that each block of the block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to the processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

<u>Conclusions Ramifications and Scope</u>

[00451] These subject technologies working alone or together may provide improved information security. DICE may boost the security of other encryptions, as well as those provided by AAC, TRIPLE and DSURF. So large-scale data theft may become less of a problem for big data handling government departments, corporates and users.

[00452] DSURF cloud 104 infrastructure 135 of Figures 19, 19A, and 19B the data security embodiments of Figures 19A and 19B, in combination operating the black box system process 1497 of Figures 41A and 42A may provide data storage that is almost impossible to breach. This is the version of these subject technologies that may be most suited to use by American consumers.

More user control of options and operations may also be provided for specialist users and applications, where securing data on the device is more important using systems according to Figure 19, and Figure 19B, in combination with the processes shown in Figures 42, and 43.

[00453] Parallel uploads and downloads of data blocks via a plurality of channels that are separate and isolated from the channel carrying their encryption key according to DICE may provide increased speed in addition to improved security. Better data security may also help reduce online financial crime and intellectual property theft.

[00454] Using two or more parallel isolated channels operating symmetrically DICE may additionally improve security and provide the bandwidth of both channels for users to allow users to operate communications with enhanced bandwidth. So that the first channel carries data encrypted by the encryption that is operating according to keys carried by the second channel; and when this is configured symmetrically: the second channel may also carry data encrypted by an encryption the details of which are carried by the first channel.

[00455] TRIPLE may also help users to enjoy more secure communications. They may also offer a security and functionality boost to users. Because their transient nature may help to make them uneconomic as well as being difficult to crack during their period of transience. A TRIPLE secured text interface could run as a web-browser extension or App on phones and computers. Operating between two users who want to send secret text to each other. The efficacy of which web-browser extension of App may be significantly boosted by the additional use of and AAC in combination with TRIPLE. Attachments may also be secured using a local version of DSURF and the keys provided using DICE. These simple examples may run via websites that run under HTTPS and other similar mathematics-based RSA type encryptions without any conflict. So that these technologies can for the most part be applied to the current infrastructures of the Internet and Cloud computing. DSURF encrypted files may also be provided via download links from a Cloud infrastructure operating according to the DSURF, and / or DICE .

[00456] TRIPL may be used to transform an intelligence asset's or agent's speech to text, then transmit the TRIPLE version over networks back to base. Where the TRIPLE is decrypted by reversing its TRIPL encryption, then reducing it back to text has the additional advantage of protecting the agent's identity. Because speech may be synthetically recreated using a generic voice at either end. Indeed, the asset's or agent's own voice may be sampled and added back to the speech at a headquarters of base of operations.

[00457] The efficacy of TRIPLE, and other forms of encryption in the current art including ones based purely on mathematics such as RSA may be improved and be made more secure by the use of statistical adjustment of their data, via an AAC prior to any other encryption that is to be used. To provide a disproportionate security boost, for very little processing power. That is considerably in excess of what is achievable by increasing processor-heavy, math-heavy complexity. As has been the paradigm for the last seventy or so years from 128k to 256k, then 512k etc. needing ever more computing resources and consuming more power to encrypt. However, quantum computing may provide the step-change in decryption capabilities that can't be sufficiently offset or mitigated by using ever-more math-heavy encryptions such as 1024k etc. Which has created the need for new approaches and is one of the main reasons these subject technologies have been created (AAC, TRIPLE, DSURF, and DICE).

[00458] The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. The previous description provides various examples of the subject technology, and the subject technology is not limited to these examples. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein but is to be accorded the full scope consistent with the language claims, wherein reference to an element in the singular is not intended to mean “one and only one” unless specifically so stated, but rather “one or more.” Unless specifically stated otherwise, the term “some” refers to one or more. Pronouns in the masculine (e.g., his) include the feminine and neuter gender (e.g., her and its) and vice versa. Headings and subheadings, if any, are used for convenience only and do not limit the invention.

[00459] Terms such as “top,” “bottom,” “front,” “rear,” “above,” “below” and the like as used in this disclosure should be understood as referring to an arbitrary frame of reference, rather than to the ordinary gravitational frame of reference. Thus, a top surface, a bottom surface, a front surface, and a rear surface may extend upwardly, downwardly, diagonally, or horizontally in a gravitational frame of reference. Similarly, an item disposed above another item may be located above or below the other item along a vertical, horizontal, or diagonal direction; and an item disposed below another item may be located below or above the other item along a vertical, horizontal, or diagonal direction.

[00460] A phrase such as an “aspect” does not imply that such aspect is essential to the subject technology or that such aspect applies to all configurations of the subject technology. A disclosure relating to an aspect may apply to all configurations, or one or more configurations. An aspect may provide one or more examples. A phrase such as an aspect may refer to one or more aspects and vice versa. A phrase such as an “embodiment” does not imply that such embodiment is essential to the subject technology or that such embodiment applies to all configurations of the subject technology. A disclosure relating to an embodiment may apply to all embodiments, or one or more embodiments. An embodiment may provide one or more examples. A phrase such an embodiment may refer to one or more embodiments and vice versa. A phrase such as a “configuration” does not imply that such configuration is essential to the subject technology or that such configuration applies to all configurations of the subject technology. A disclosure relating to a configuration may apply to all configurations, or one or more configurations. A configuration may provide one or more examples. A phrase such as configuration may refer to one or more configurations and vice versa.

[00461] The word “exemplary” is used herein to mean “serving as an example or illustration.” Any aspect or design described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects or designs.

[00462] All structural and functional equivalents to the elements of the various aspects described throughout this disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims. Moreover, nothing disclosed herein is intended to be dedicated to the public regardless of whether such disclosure is explicitly recited in the claims. No claim element is to be construed under the provisions of 35 U.S.C. §112, sixth paragraph, unless the element is expressly recited using the phrase “means for” or, in the case of a method claim, the element is recited using the phrase “step for.” Furthermore, to the extent that the term “include,” “have,” or the like is used in the description or the claims, such term is intended to be inclusive in a manner similar to the term “comprise” as “comprise” is interpreted when employed as a transitional word in a claim.

CLAIMS

What is claimed is:

1. An encryption system for digital communications, comprising:

a first device coupled to a telecommunications network through a first network connection, wherein the first device is configured to operate on a first channel to transmit and receive data;

a second device coupled to the telecommunications network through a second network connection, wherein the second device is configured to operate on the first channel to transmit and receive data;

a third network connection configured to connect the first device to a second channel in the telecommunications network;

a fourth network connection configured to connect the second device to the second channel in the telecommunications network, wherein:

the first channel is isolated and separate from the second channel,

encrypted data passes between the first device and the second device through the first channel, and

decryption data configured to decrypt the encrypted data in the first channel, passes between the first device and the second device through the second channel.

2. The system of claim 1, further comprising a software application resident on the first device, wherein the software application is configured to connect the first device to the second channel through the third network connection.

3. The system of claim 1, wherein:

encrypted data passes between the first device and the second device through the second channel, and

decryption data configured to decrypt the encrypted data in the second channel, passes between the first device and the second device through the first channel.

4. The system of claim 1, further comprising a SIM apparatus, an e-SIM, or a virtual SIM on either the first device or the second device, wherein the SIM apparatus, e-SIM, or virtual SIM is configured to connect to the telecommunications network and to provide communications through either the first channel or the second channel.

5. The system of claim 1, further comprising a router, wherein one of the network connections that provides communications through one of the channels is connected to the router.

6. The system of claim 1, further comprising an antenna, wherein one of the telecommunications network connections that provides communications through one of the channels is connected to the antenna.

7. The system of claim 1, wherein one of the network connections that provides communications through one of the channels is connected to a cellular network.

8. The system of claim 1, wherein:

either the first device or the second device is configured to execute a set of instructions, and in executing those instructions a file is divided into blocks of data,

the blocks of data are randomly reordered and recorded into a non-transient file storage system,

the locations of which randomly reordered blocks of data is recorded sequentially into an array of addressable pointers,

a copy of the randomly reordered non-transient file storage is sent from the first device to the second device through the either the first channel or the second channel, and

a copy of the array of addressable pointers is sent from the first device to the second device through the other channel.

9. The system of claim 1, wherein either the first device, the second device, or a connected computing device executes a set of instructions configured to encrypt the data according to a mathematically based encryption method.

10. The system of claim 1, wherein either the first device, the second device, or a connected computing device executes a set of instructions configured to encrypt the encrypted data according to a logical cipher-based encryption method using a random seed, wherein the random seed is an alphanumeric character or data pattern.

11. The system of claim 1, wherein either the first device, the second device, or a connected computing device executes a set of instructions configured to encrypt the encrypted data according to a logical cipher-based encryption method of adjusting a relative frequency of occurrence of an alphanumeric character or a data pattern that is sent through a coupled telecommunications

channel.

12. The system of claim 11, wherein a wildcard comprising disinformation is substituted for the alphanumeric character or the data pattern.

13. The system of claim 1, wherein either the first device, the second device, or a connected computing device executes a set of instructions configured to encrypt the encrypted data according to an encryption method that randomizes a timed periodicity parameter for the use and replacement of a paired encryption and key.

14. A method of encrypting digital communications, comprising:

establishing a first channel to transmit and receive data through a first network connection of a first device coupled to a telecommunications network;

connecting a second device to the first channel through a second network connection coupled to the telecommunications network;

establishing a second channel in the telecommunications network, wherein the first device is connected to the telecommunications network via a third network connection;

establishing a fourth network connection configured to connect the second device to the second channel in the telecommunications network, wherein:

the first channel is isolated and separate from the second channel,

encrypted data passes between the first device and the second device through the first channel, and decryption data configured to decrypt the encrypted data in the first channel, passes between the first device and the second device through the second channel.

15. The method of claim 14, further comprising executing a set of instructions by either the first device or the second device, configured to:

divide a file into blocks of data;

randomly reorder and record the blocks of data into a non-transient file storage system;

sequentially record into an array of addressable pointers, the locations of the randomly reordered blocks of data;

send a copy of the randomly reordered non-transient file storage from the first device to the second device through either the first channel or the second channel; and

send a copy of the array of addressable pointers from the first device to the second device through the other channel.

16. The method of claim 14, further comprising executing a set of instructions by either the first device, the second device, or a connected computing device, configured to encrypt the encrypted data according to a mathematically based encryption method.

17. The method of claim 14, further comprising executing a set of instructions by either the first device, the second device, or a connected computing device, configured to encrypt the encrypted data according to logical cipher-based encryption method using a random seed, wherein the random seed is an alphanumeric character or data pattern.

18. The method of claim 14, further comprising executing a set of instructions by either the first device, the second device, or a connected computing device, configured encrypt the encrypted data according to an encryption method that randomizes a timed periodicity parameter for the use and replacement of a paired encryption and key.

19. The method of claim 14, further comprising:

passing encrypted data between the first device and the second device through the second channel; and

passing decryption data configured to decrypt the encrypted data in the second channel, between the first device and the second device through the first channel.

20. A computer program product for encrypting digital communications, the computer program product comprising a computer readable storage medium having program instructions embodied therewith, wherein an execution of the program instructions cause a processor to:

establish a first channel to transmit and receive data through a first network connection of a first device coupled to a telecommunications network;

connect a second device to the first channel through a second network connection coupled to the telecommunications network;

establish a second channel in the telecommunications network, wherein the first device is connected to the telecommunications network via a third network connection;

establish a fourth network connection configured to connect the second device to the second channel in the telecommunications network, wherein:

the first channel is isolated and separate from the second channel, encrypted data passes between the first device and the second device through the first channel, and

decrypt data configured to decrypt the encrypted data in the first channel, passes between the first device and the second device through the second channel.

21. The computing program product of claim 20, wherein the instructions cause the processor to perform further acts comprising:

dividing a file into blocks of data;

randomly reordering and recording the blocks of data into a non-transient file storage system;

sequentially recording into an array of addressable pointers, the locations of the randomly reordered blocks of data;

sending a copy of the randomly reordered non-transient file storage from the first device to the second device through either the first channel or the second channel; and

sending a copy of the array of addressable pointers from the first device to the second device through the other channel.

22. The computing program product of claim 20, wherein the instructions cause the processor to perform further acts comprising executing a set of instructions by either the first device, the second device, or a connected computing device, configured to encrypt the encrypted data according to a mathematically based encryption method.

23. The computing program product of claim 20, wherein the instructions cause the processor

to perform further acts comprising encrypting the encrypted data according to logical cipher-based encryption method using a random seed, wherein the random seed is an alphanumeric character or data pattern.

24. The computing program product of claim 20, wherein the instructions cause the processor to perform further acts comprising executing a set of instructions by either the first device, the second device, or a connected computing device, configured encrypt the encrypted data according to an encryption method that randomizes a timed periodicity parameter for the use and replacement of a paired encryption and key.

25. The computing program product of claim 20, wherein the instructions cause the processor to perform further acts comprising:

passing encrypted data between the first device and the second device through the second channel; and

passing decryption data configured to decrypt the encrypted data in the second channel, between the first device and the second device through the first channel.